Welcome
United States Patent and Trademark Office

Image indexing using weighted color histogram

Du-Sik Park Jong-Seung Park Tae Yong Kim Joon Hee Han
Signal Process. Lab., Samsung AIT, Yongin, South Korea;

This paper appears in: **Image Analysis and Processing, 1999. Proceeding International Conference on**

Meeting Date: 09/27/1999 - 09/29/1999
Publication Date: 27-29 Sept. 1999
Location: Venice Italy
On page(s): 909 - 914
Reference Cited: 10
Number of Pages: xxii+1232
Inspec Accession Number: 6489047

Abstract:
Image **indexing** is the process of image retrieval from databases of images o based on their contents. Specifically **histogram**-based algorithms are conside effective for **color** image **indexing**. We suggest a new method of **color** space quantization in the CIELUV **color** space, named weighted LUV quantization. W method, each bin in the LUV space has a different weighting factor, which is a the **histogram** intersection. The weighted LUV **histogram** intersection provid advantage of perceptual uniformity of the CIELUV **color** space. An additional a is the consideration of perceptual sensitivity to more saturated **colors** by the weighting factor

Index Terms:
content-based retrieval database **indexing** image colour analysis quantisation (signa analysis visual databases CIELUV **color** space bin weighting **color** space quantiza contents **histogram** intersection **histogram**-based algorithms image databases ima **indexing** image retrieval perceptual sensitivity perceptual uniformity saturated **colc** databases weighted LUV quantization weighted **color histogram**

Documents that cite this document
There are no citing documents available in IEEE Xplore at this time.

Image Indexing using Weighted Color Histogram*

Du-Sik Park[†§], Jong-Seung Park[‡], TaeYong Kim[§] and Joon Hee Han[§]

[†] Signal Processing Lab., Samsung A.I.T., Yongin, Korea
[‡] Human Computing Research Dept., ETRI, Taejon, Korea
[§] Department of Computer Science and Engineering, POSTECH
San 31, Hyo-Ja-Dong, Pohang, 790-784, Korea
{pds@, parkjs@falcon., kimty@hawk., joonhan@}postech.ac.kr

Abstract

Image indexing is the process of image retrieval from databases of images or videos based on their contents. Specifically, histogram-based algorithms are considered to be effective for color image indexing. We suggest a new method of color space quantization in the CIELUV color space, named weighted LUV quantization. With this method, each bin in the LUV space has a different weighting factor, which is applied to the histogram intersection. The weighted LUV histogram intersection provides the advantage of perceptual uniformity of the CIELUV color space. An additional advantage is the consideration of perceptual sensitivity to more saturated colors by the use of a weighting factor.

1. Introduction

The image indexing process must satisfy the automated extraction of features, efficient indexing and the effective retrieval of images within a database. Color features have proven to be efficient in discriminating between relevant and non-relevant images. Moreover, histogram-based techniques have been widely studied and are considered to be effective for color image indexing [9]. The key issue of histogram-based techniques is the selection of an appropriate color space and the quantization of the selected color space.

In this paper, we suggest a quantization and histogram matching algorithm in the CIELUV color space. In the proposed quantization, the LUV space, surrounding all of the CIELUV color space defined by conversion of RGB color space, is subdivided into uniform sized bins along with each axis. Each bin in the LUV space has a different weighting factor according to the volume of the CIELUV color space included within it. We denote this quantization scheme as weighted LUV quantization. The weighting factor for each bin is applied to the calculation of the histogram intersection to measure the similarity between a query image and database images. This weighted histogram comparison algorithm is similar to those of Swain and Ballard [7], Stricker and Swain [6], and Funt and Finlayson [3] except the concept of weight. Using this weighting scheme, we remove a number of bins that are almost useless for discriminating images in the view of perceptivity. We show the plausible performance of our algorithm for image indexing through experiments.

*This work was supported in part by the Automation Research Center (ARC) designated by KOSEF.

2. Image indexing

In this section, we review the histogram comparisons and the quantization of the selected color space for image indexing.

2.1. Histogram-based indexing

The advantage of using color histograms is their robustness with respect to geometric changes of projected objects. Histograms are invariant to translation and rotation around the viewing axis and vary slowly with changes of view angle, scale, and occlusion. The colors in an image are mapped into a discrete color space containing n colors. A histogram of image I is an n-dimensional vector, where each element represents the number of pixels of color j in image I. Each element of a histogram is normalized so that the histogram represents the image without regard to the image size. The element of the normalized histogram $H(I)$

is defined as

$$H_j(I) = \tilde{H}_j(I) / \sum_{j=1}^{n} \tilde{H}_j(I) , \tag{1}$$

where $\tilde{H}_j(I)$ is the number of pixels of color j in image I. We denote $d(H(I), H(I'))$ as the distance between two histograms.

Swain and Ballard [7] introduced a histogram matching method called ***histogram intersection***. Given a pair of histograms, $H(I)$ and $H(I')$, of image I and image I', respectively, each containing n bins, they defined the histogram intersection as follows:

$$\tilde{H}(I) \cap \tilde{H}(I') = \frac{\sum_{j=1}^{n} \min(\tilde{H}_j(I), \tilde{H}_j(I'))}{\sum_{j=1}^{n} \tilde{H}_j(I)} . \tag{2}$$

The denominator normalizes the histogram intersection and makes the value of the histogram intersection between 0 and 1. The measure $\tilde{H}(I) \cap \tilde{H}(I')$ is analogous to the L_1-norm,

$$d_{L_1}(\tilde{H}(I), \tilde{H}(I')) = \sum_{j=1}^{n} |H_j(I) - H_j(I')| , \tag{3}$$

as defined by Swain and Ballard [7], Stricker and Swain [6], and Funt and Finlayson [3]. For a given distance T, two histograms are said to be ***similar*** if their distance is less than or equal to T, and an image in a database is retrieved in response to the given query image.

2.2. Quantization

The histogram dimension (the number of histogram bins) n is determined by a color representation scheme and quantization level. Most color spaces represent a color as a 3D vector with real values (e.g. RGB, CIE XYZ, HSV, CIELUV). We quantize the color space of three axes into k bins for the first axis, l bins for the second axis and m bins for the third axis. The histogram can be represented as an n-dimensional vector where $n = k \times l \times m$.

In general high resolution schemes, the histogram of RGB color space with $[255, 255, 255]$ range of three axes is represented as a 2^{24}-dimensional vector, HSV color space with $[360, 100, 100]$ range of three axes as a 3600000-dimensional vector, CIELUV space with $[100, 354, 262]$ range of three axes as a 9274800-dimensional vector. These high resolution representations, however, are unnecessarily large for image indexing. Because the retrieval performance is saturated when the number of bins is increased beyond some value, normalized color histogram difference was a satisfactory measure of frame dissimilarity, even when colors were quantized into only 128 bins (8 green by 8 red by 4 blue) [4, 9].

3. Color spaces

The word "color" may be interpreted in several ways: a certain kind of light, its effect on the human eye, or most important of all, the result of this effect in the mind of the viewer [2]. Color is the perceptual result of visible light, which lies within the range of approximately 380nm - 750nm of the spectrum wavelength, incident upon the retina.

3.1. CIE XYZ color space

The assumption that there are three types of cone receptors in the retina is widely accepted, so three components are necessary and sufficient to describe a color. Accordingly, the set of all perceivable colors can be represented within a three-dimensional space.

The axes of a color space, called primary colors, can be chosen arbitrarily. A convenient set, universally used for color measurement, is the CIE 1931 (X,Y,Z)-system adopted by the *Commission Internationale de l'Éclairage (CIE)* [10]. Each distinct point in the CIE XYZ space corresponds to a unique color perception. In this space, the pure color component in the absence of brightness, such as hue and chroma, can be represented with x and y chromaticity coordinates defined by

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}. \tag{4}$$

3.2. RGB color space

RGB color space is represented with red (R), green (G), and blue (B) primaries and is an additive system. An additive RGB system is specified by the chromaticities of its primaries and its white point. This system's extent (gamut) is given in the (x,y) chromaticity diagram [10] by a triangle whose vertices are the chromaticities of the primaries.

RGB values in a particular set of primaries can be transformed to and from CIE XYZ by three-by-three matrix transform. To transform from RGB into CIE XYZ the following transform is used [5]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.607 & 0.174 & 0.2000 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.116 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} . \tag{5}$$

Because white is normalized to unity, the middle row sums to unity. To recover the white point, we need to transform $RGB = [1, 1, 1]$ to CIE XYZ before computing x and y.

The RGB color space provides a simple and fast computation. However, it is neither the perceptually uniform space nor the intuitive space.

3.3. HSV color space

The hue (H), saturation (S), and value (V) system is based on a warped version of the RGB space and is directly related to intuitive color notions of hue, saturation and brightness.

The conversion from the RGB color space to the HSV color space is done through the following equation [1]:

$$h' = \begin{cases} \frac{(g-b)}{\delta} & \text{if } r = max \\ \frac{2+(b-r)}{\delta} & \text{if } g = max \\ \frac{4+(r-g)}{\delta} & \text{if } b = max \end{cases} \quad (6)$$

$$v = max$$
$$s = \frac{max - min}{max}$$
$$h = h' * 60, \quad (7)$$

where $max = MAX(r, g, b)$, $min = MIN(r, g, b)$, $\delta = max - min$, and (h, s, v) is the corresponding point of (r, g, b) in the HSV color space. For $r, g, b \in [0 \cdots 1]$, the conversion gives $h \in [0 \cdots 360]$ and $s, v \in [0 \cdots 1]$.

The intuitiveness of the HSV color space is very useful because we can quantize each axis independently. Wan and Kuo [9] reported that a color quantization scheme based on HSV color space performed much better than one based on RGB color space. Because HSV color space involves different computations around 60 degree segments of the hue circle, the visible discontinuities occur in the color space. HSV color space linearly converted from RGB color space is also not a perceptually uniform space.

3.4. CIELUV color space

Consider the distance from color $C_1 = (X_1, Y_1, Z_1)$ to color $C_1 + \Delta C$, and the distance from color $C_2 = (X_2, Y_2, Z_2)$ to color $C_2 + \Delta C$, where $\Delta C = (\Delta X, \Delta Y, \Delta Z)$. Both distances are equal to ΔC, yet in general they will not be perceived as being equal [1]. Perceptual uniformity means that the same ΔC at two different points in the color space makes the equal perceivable color difference. CIE XYZ, RGB, and HSV color spaces do not exhibit perceptual uniformity. There are two perceptually uniform color spaces that are the agreed standards in CIE. One of the two is CIELUV color space. The three variables L^*, u^*, and v^* are defined by [10]:

$$L^* = \begin{cases} 116(\frac{Y}{Y_n})^{\frac{1}{3}} - 16 & \text{if } \frac{Y}{Y_n} > 0.008856 \\ 903.3\frac{Y}{Y_n} & \text{otherwise} \end{cases} \quad (8)$$

$$u^* = 13L^*(u' - u'_n)$$
$$v^* = 13L^*(v' - v'_n) \quad , \quad (9)$$

where u', v' and u'_n, v'_n are calculated from,

$$u' = \frac{4X}{X+15Y+3Z}, \; v' = \frac{9Y}{X+15Y+3Z}$$
$$u'_n = \frac{4X_n}{X_n+15Y_n+3Z_n}, \; v'_n = \frac{9Y_n}{X_n+15Y_n+3Z_n} \quad . \quad (10)$$

The tristimulus values X_n, Y_n and Z_n are those of the illuminant, with Y_n equal to 1. The total color difference ΔE^*_{uv} between two colors in CIELUV color space is calculated from:

$$\Delta E^*_{uv} = [(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2]^{\frac{1}{2}} \; , \quad (11)$$

where ΔL^*, Δu^* and Δv^* are the difference in L^*, u^* and v^*, between two colors, respectively.

4. Weighted quantization in CIELUV space

In uniform quantization, each axis of the color space is uniformly subdivided into a prespecified number of bins and each bin is the same size. The advantage of uniform splitting method is that it is straightforward and simple [9]. However, the disadvantage of uniform quantization for perceptually non-uniform color spaces, such as RGB and HSV spaces, is that they don't take into account the perceptual similarity between different bins. To overcome this perceptual dissimilarity problem, we select the CIELUV color space, which is the same color space that Taycher [8] chose for uniform quantization.

Figure 1 shows RGB space and CIELUV color spaces surrounded by LUV space. LUV space is represented by the uniformly quantized lattices as shown in Figure 1(b) and (c). Each bin in the LUV space includes a different volume of the CIELUV color space defined by a two step transformation of the RGB space using Eqs.(5), (8) and (9).

Here, we introduce the weighted quantization method in LUV space. In this method, each bin in LUV space has a different weighting factor according to the volume of CIELUV color space included within it. We denote this method as weighted LUV quantization. The weighting factor for each bin is applied to the calculation of the histogram intersection in the form of

$$\left[\tilde{H}(I) \cap \tilde{H}(I')\right]_w = \frac{\sum_{j=1}^{n} W_j \min(\tilde{H}_j(I), \tilde{H}_j(I'))}{\sum_{j=1}^{n} W_j \tilde{H}_j(I)} \; , \quad (12)$$

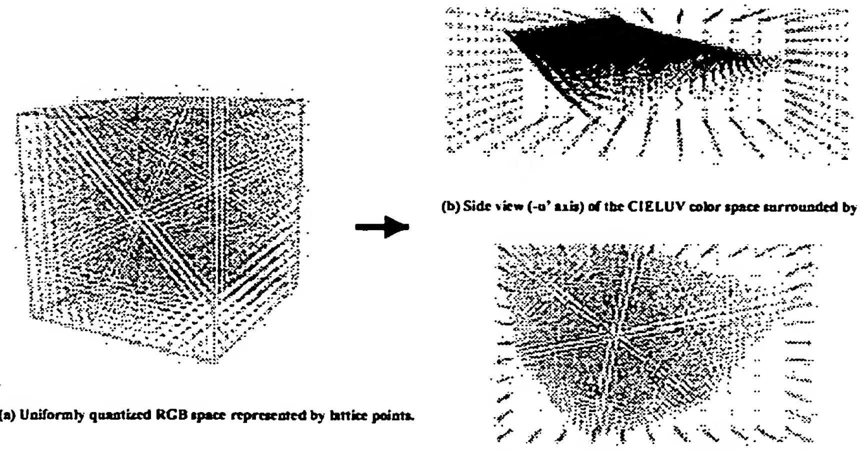
where W_j is the weight for the bin j in the LUV space. We denote this histogram intersection, with subscript w, $\left[\tilde{H}(I) \cap \tilde{H}(I')\right]_w$, as the weighted histogram intersection.

The weight for each bin is determined by the following procedures:

1) Transform R, G, and B values at the lattice points in a uniformly subdivided RGB color space with a prespecified number into L^*, u^*, and v^* values of the CIELUV color space.

2) Compute the probability as an estimated volume included in each bin:

$$P_i = \frac{N_i}{\sum_{j=1}^{n} N_j}, \quad i = 1, 2, ..., n, \quad (13)$$



(a) Uniformly quantized RGB space represented by lattice points.

(b) Side view (-u' axis) of the CIELUV color space surrounded by the LUV space.

(c) Top view (+z axis) of the CIELUV color space surrounded by the LUV space.

Figure 1. RGB space and a perspective view of the CIELUV color space surrounded by LUV space.

where N_i is the pixel number in bin i, n is the total number of bins in the space, and P_i is the probability that a pixel will fall in bin i.

3) Compute the weight,

$$W_i = \begin{cases} 0 & \text{if } P_i = 0, \\ \frac{\frac{1}{P_i}}{\sum_{j=1}^{n} \frac{1}{P_j}} & \text{otherwise} \end{cases} \qquad (14)$$

where W_i is the weight for bin i. The weight of a bin, which includes the smaller volume within it, will be the larger. That is, the bin including highly saturated colors will get the larger weight.

Weighted LUV quantization is designed to provide the advantage of perceptual uniformity in the histogram similarity comparison. Using this weighting scheme, bins that do not reflect the perceptual discrimination are eliminated, and the sensitivity of a highly saturated color is considered to be important by the weight.

5. Experimental results

Experiments were carried out to present the color indexing performance of weighted LUV quantization. To compare the performance, the histogram intersection is calculated in the HSV space and the LUV space with the typical uniform quantization and weighted quantization. In computing the weighting factor, we have examined sufficient samples for the probability P_i in order to be proportional to the volume of the CIELUV color space included within a bin.

Captured frames from "Star TV" and the movie "Deep Impact" were used as database images. We denote the Star TV images as the first database set and the Deep Impact images as the second database set. The first database set has 47 images and the second database set has 63 images. A query image was selected from a database set and the histogram of the image was compared with the histograms of the rest of the images in the database set.

Figure 2 shows the result when color indexing for the first database set is applied to the quantization scheme for the each space. The first two columns in Figure 2 show the sequence of the original images and the next three columns display the sequence ranked as the top ten similar images using three quantization schemes: uniform quantization of HSV space, typical uniform quantization of LUV space (LUV), and weighted LUV quantization (LUVw). The lowermost image of each quantization scheme is the query image followed by nine matched images in the order of similarity. The numbers on the bottom of each image are the sequence number in the original scene and the value of normalized similarity to the query image (italicized number).

Table 1 shows the variance (VAR) of the three relevant images (including query image and the two top ranked images) and the percentile of the difference (AD(%)) between the average histogram similarity of the three relevant images and the average histogram similarity of the seven non-relevant images. Q. Step means the quantization step used. The adaptive quantization step has the $15 \times 9 \times 9$ step for the HSV space and the $7 \times 13 \times 13$ step for the LUV space, which produces the desirable result as described by Wan and Kuo [9].

Figure 3 shows the result of color indexing for the second database set. The quantization step used is the Adaptive for HSV and LUV and the $5 \times 5 \times 5$ step for LUVw.

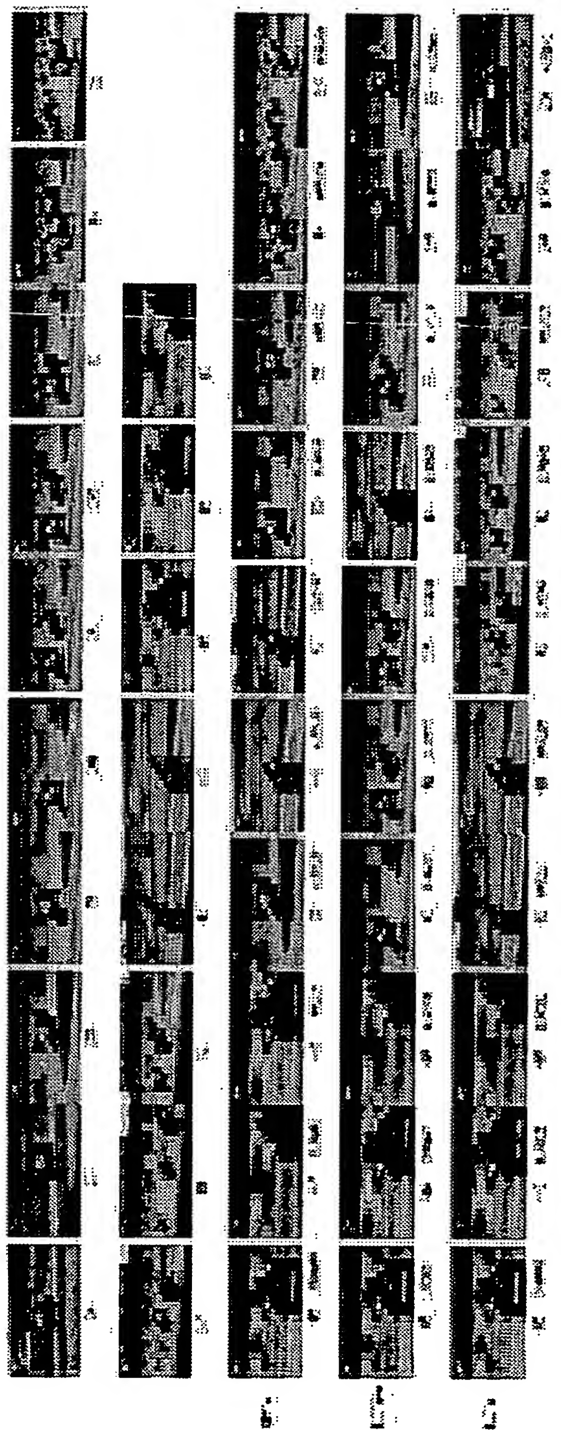


Figure 2. Comparison of the uniform quantization in HSV space and LUV space for the first database set.

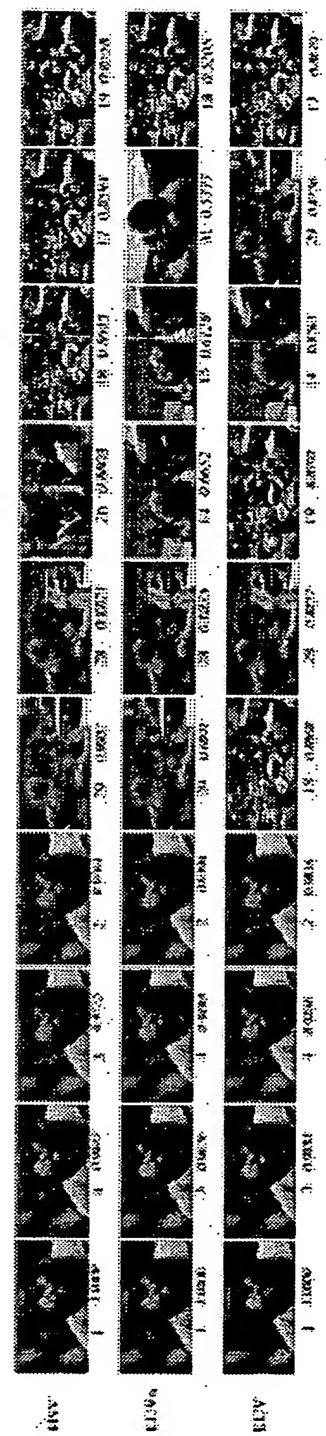


Figure 3. Comparison of uniform quantization in the HSV space and the LUV space for the second database set.

Table 1. Comparison of the variance and the percentile of the difference of the average similarity for the first database set.

Q. Step	Measure	HSV	LUVw	LUV
$3 \times 3 \times 3$	VAR	0.0006	0.0087	0.00013
	AD(%)	7.60	39.23	6.29
$5 \times 5 \times 5$	VAR	0.0016	0.0012	0.00052
	AD(%)	10.27	60.02	5.20
Adaptive	VAR	0.0024	0.0032	0.00068
	AD(%)	14.64	57.51	11.01

Table 2. Comparison of variance and the percentile of the difference of the average similarity for the second database set.

Q. Step	Measure	HSV	LUVw	LUV
$5 \times 5 \times 5$	VAR	0.00012	0.0007	0.00004
	AD(%)	11.08	33.80	10.84
Adaptive	VAR	0.00036	0.0017	0.00006
	AD(%)	17.98	34.69	11.12

Table 2 shows the variance (VAR) of the four relevant images and the percentile of the difference (AD(%)) of the average histogram similarity between the four relevant images and the six non-relevant images for the second database set.

In all quantization steps, the uniform LUV quantization scheme has the least variance for the relevant images and the weighted LUV quantization scheme has the maximum difference of the average similarity between the relevant images and the non-relevant images. For image indexing, a large difference of average similarity and a small variance are useful characteristics for discriminating between relevant images and non-relevant images. From the viewpoint of this discussion, the weighted LUV quantization scheme shows plausible discrimination performance. Even in the low resolution quantization step, $5 \times 5 \times 5$, this proposed method presents the best performance in discriminating characteristics, with an appropriately small variance.

6. Conclusion

To improve color indexing performance, the weighted LUV quantization scheme has been proposed. For given database sets, experimental results show that the weighted LUV quantization method, even for low resolution quantization, gives better performance than others.

The proposed weighted LUV quantization has a complete three dimensional subdivision scheme. To speed up the processing, we can apply a pseudo three dimensional subdivision scheme in which three axes are quantized independently of each other. Another interesting issue is to find the method assigning the weighting factor to special bins for indexing images having special color, such as flesh tone. These remain as further investigation issues.

References

[1] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley Publishing, 1996.

[2] J. Fred W. Billmeyer and M. Saltzman. *Principles of Color Technology.* John Wiley & Sons, 1981.

[3] B. V. Funt and G. D. Finlayson. Color constant color indexing. *PAMI*, 17(5):522–529, May 1995.

[4] J. R. Kender and B.-L. Teo. Video scene segmentation via continuous video colors. In *CVPR*, pages 367–373, 1998.

[5] W. N. Sproson. *Colour science in television and display systems.* Adam Hilger, 1983.

[6] M. Stricker and M. Swain. The capacity of color histogram indexing. In *CVPR94*, pages 704–708, 1994.

[7] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[8] L. Taycher. *Image Feature Extraction Subsystem of the ImageRover WWW Image Search System.* Mater thesis, Boston University, 1997.

[9] X. Wan and C.-C. J. Kuo. Color distribution analysis and quantization for image retrieval. In *SPIE proceedings, vol. 2670*, February, 1996.

[10] G. Wyszecki and W. S. Stiles. *Color Science, concept and Methods, Quantitative Data and Formulae.* John Wiley & Sons, 1982.

Efficient color histogram indexing for quadratic forı distance functions

Hafner, J. Sawhney, H.S. Equitz, W. Flickner, M. Niblack, W.
IBM Almaden Res. Center, San Jose, CA, USA;

This paper appears in: **Pattern Analysis and Machine Intelligence, IEEE Transactions on**

Publication Date: July 1995
On page(s): 729 - 736
Volume: 17 , Issue: 7
ISSN: 0162-8828
Reference Cited: 20
CODEN: ITPIDJ
Inspec Accession Number: 5000965

Abstract:
In image retrieval based on **color**, the weighted distance between **color histo** two images, represented as a quadratic form, may be defined as a match mea However, this distance measure is computationally expensive and it operates dimensional features (O(N)). We propose the use of low-dimensional, simple t distance measures between the **color** distributions, and show that these are l bounds on the **histogram** distance measure. Results on **color histogram** ma large image databases show that prefiltering with the simpler distance measu significantly less time complexity because the quadratic **histogram** distance i computed on a smaller set of images. The low-dimensional distance measure used for **indexing** into the database

Index Terms:
computational complexity image colour analysis image matching **indexing** informat query processing visual databases **color** based image retrieval **color** distributions c **histogram indexing** feature matching image databases image querying low dimen measure lower bounds match measure quadratic form distance functions time comp

Documents that cite this document
Select link to view other documents in the database that cite this one.

Reference list:

1, R. Agrawal,, C. Faloutsos , and A. Swami,, "Efficient similarity search in se databases," *Fourth Int'l Conf. Foundations of Data Organization and Algorithm* Ill., 1993.
[Buy Via Ask*IEEE]

2, E. Binaghi,, I. Gagliardi, and R. Schettini,, "Indexing and fuzzy logic-based color images," *Visual Database Systems, II, IFIP Trans. A-7,* pp. 79-92, Elsevi Publishers, 1992.
[Buy Via Ask*IEEE]

3, T.M. Cover and J.A. Thomas,, *Elements of Information Theory.* Wiley Serie Telecommunications, John Wiley & Sons, 1991.
[Buy Via Ask*IEEE]

4, P. Dewilde and E.F. Deprettere,, "Singular value decomposition, an introdu *and Signal Processing: Algorithms, Applications, and Architectures,* E.F. Depre pp. 3-42.New York: Elsevier Science Publishing Co., 1988.
[Buy Via Ask*IEEE]

5, T. Feder,, "On a quadratic form for points in space," IBM Internal Note, 19

6, G.H. Golub and C.F. van Loan,, *Matrix Computations.* The Johns Hopkins U 1989.
[Buy Via Ask*IEEE]

7, J. Hasegawa,, N. Okada, and J. Toriwaki,, "Intelligent retrieval of chest x-r database using sketches," *Systems and Computers in Japan,* vol. 20, no. 7, pp 1989.
[Buy Via Ask*IEEE]

8, K. Hirata and T. Kato. , "Query by visual example," *Adv. in Database Techn EDBT '92, Third Int'l Conf. on Extending Database Technology.*Vienna, Austria Verlag, Mar. 1992.
[Buy Via Ask*IEEE]

9, IMSL,, Inc.,, *User's Manual MATH/Library.*Texas: IMSL, Inc., 1991.
[Buy Via Ask*IEEE]

10, M. Ioka,, "A method of defining the similarity of images on the basis of co information," *Technical report RT-0030,* IBM Tokyo Research Lab, 1989.

11, M.A. Ireton and C.S. Xydeas,, "Classification of shape for content retrieva in a multimedia database," *Sixth Int'l Conf. on Dig. Proc. of Signals in Comm.* 116,Loughborough, U.K., Sept. 2-6, 1990.
[Abstract] [PDF Full-Text (388KB)]

12, H.V. Jagadish,, "A retrieval technique for similar shapes," *Int'l Conf. on M of Data, SIGMOD 91,* pp. 208-217,Denver, Colo., May 1991.
[Buy Via Ask*IEEE]

13, D.G. Luenberger,, *Linear and Nonlinear Optimization.* Mass.: Addison-We
[Buy Via Ask*IEEE]

14, M. Miyahara and Y. Yoshida,, "Mathematical transform of (R, G, B) color d Munsell (H, V, C) color data," *Visual Communication and Image Processing,* vo pp. 650-657, SPIE, 1988.
[Buy Via Ask*IEEE]

15, A. Mood,, F. Graybill, and D. Boes,, *Introduction to the Theory of Statistic* Hill series in probability and statistics, McGraw-Hill, third edition, 1963.
[Buy Via Ask*IEEE]

16, W. Niblack,, R. Berber,, W. Equitz,, M. Flickner,, E. Glasman,, D. Petkovic Yanker,, "The QBIC project: Querying ims. by content using color, texture, an *SPIE 1908, Storage and Retrieval for Image and Video Dbases,* Feb. 1993.
[Buy Via Ask*IEEE]

17, C.L. Novak and S.A. Shafer,, "Color vision," *Physics-Based Vision COLOR,* Jones and Bartlett, 1992.
[Buy Via Ask*IEEE]

18, M. Otterman,, "Approximate matching with high dimensionality R-trees," scholarly paper, Dept. of Computer Science, Univ. of Maryland, College Park, M supervised by C. Faloutsos, 1992.

19, A. Pentland,, R. Picard,, G. Davenport, and B. Welsh,, "The BT/MIT projec advanced image tools for telecommunications: An overview," *Technical report* Media Lab. Perceptual Comp. Group TR, 1992.

20, M.J. Swain and D.H. Ballard,, "Color indexing," *Int'l J. Computer Vision,* v pp. 11-32, 1991.
[Buy Via Ask*IEEE]

100×100 gray scale image with 256 gray levels. Table I shows the average processing time and computational complexity for Wang and Pavlidis' method [1] and the proposed method 2. As shown in Table I, in case of extracting topographic features using the method 2, the processing time was approximately five times as fast as the Wang and Pavlidis' method.

TABLE I
AVERAGE PROCESSING TIME AND COMPUTATIONAL COMPLEXITY FOR EACH METHOD

Method	Average processing time	Computational complexity
Wang and Pavlidis' method	3.82 sec	$i \times j \times (8 \times M^2 + C)$
Proposed method 2	0.76 sec	$i \times j \times (4 \times N^2 + C)$

i : the width of input image
j : the height of input image
M : neighborhood size for calculating the first and second partial derivatives
N : neighborhood size for convolution with Laplacian-like masks
C : the number of comparative operations

V. CONCLUDING REMARKS

In this paper, we proposed a new method for extracting topographic features directly from a gray scale character image without calculating eigenvalues and eigenvectors of the underlying image intensity surface.

For real world character images, the gradient magnitude would be rarely zero at the center of pixel. Therefore, if the Wang and Pavlidis' method is used for topographic feature extraction, the eigenvectors must be approximated mostly by two perpendicular directions, and the calculation to get approximated directions requires much reluctant efforts. In addition to that, by employing such an approximation for the eigenvectors, unnecessary ridges and peaks may be extracted at the places such as bend points, starting points, end points, or corresponding hillside.

In case of extracting topographic features using the proposed method, the processing time was approximately five times as fast as the Wang and Pavlidis' method because only simple comparative operations were used, and unnecessary topographic features need not be extracted at bend points, starting points, and end points by taking the local information of gray scale character image into account in determining principal orthogonal elements. In addition to that, experimental results with Levi and Montanari's data revealed that the proposed method was also very effective for gray scale skeletonization for character recognition.

ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their helpful comments in improving the earlier draft of this paper. This research was supported by the 1992 Directed Basic Research Fund of Korea Science and Engineering Foundation.

REFERENCES

[1] L. Wang and T. Pavlidis, "Direct gray scale extraction of features for character recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1,053-1,067, Oct. 1993.

[2] T. Pavlidis and Wolberg, "An algorithm for the segmentation of bilevel images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 570-575, Miami, June 1986.

[3] H. Murakami and B.V.K.V. Kumar, "Efficient calculation of primary images from a set of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, pp. 511-515, 1982.

[4] C.M. Leung, "A practical basis set for Chinese character recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 532-537, San Francisco, June 1985.

[5] G. Levi and U. Montanari, "A gray-weighted skeleton," *Information and Control*, vol. 17, pp. 62-91, 1970.

[6] S.W. Lam, A.C. Girardin, and S.N. Srihari, "Gray scale character recognition using boundary features," *Proc. SPIE Conf. Machine Vision Applications in Character Recognition and Industrial Inspection*, vol. 1,661, pp. 98-105, San Jose, 1992.

[7] T. Pavlidis, "Recognition of printed text under realistic conditions," *Proc. Second IPTP Conf.*, pp. 65-76, Tokyo, Jan. 1992.

[8] T. Pavlidis, L. Wang, J. Zhou, W.J. Sakoda, and J. Rocha, "Recognition of poorly printed text by direct extraction of features from gray scale," *Proc. SPIE Conf. Machine Vision Applications in Character Recognition and Industrial Inspection*, vol. 1,661, pp. 1-9, San Jose, Feb. 1992.

[9] L. Wang and T. Pavlidis, "A geometric approach to machine-printed character recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 665-668, Champaign, Ill., June 1992.

[10] R.M. Haralick, L.T. Watson, and T.J. Laffey, "The topographic primal sketch," *Int'l J. Robotics Research*, vol. 2, pp. 50-72, 1983.

[11] D.H. Kim, Y.S. Hwang, S.T. Park, E.J. Kim, S.H. Paek, and S.Y. Bang, "Handwritten Korean character image database PE92," *Proc. Second Int'l Conf. Document Analysis and Recognition*, pp. 470-473, Tsukuba Science City, Japan, Oct. 1993.

[12] L. Lam, S.-W. Lee, and C.Y. Suen, "Thinning methodologies—a comprehensive survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869-885, Sept. 1992.

[13] S.-W. Lee, L. Lam, and C.Y. Suen, "A systematic evaluation of skeletonization algorithms," *Pattern Recognition and Artificial Intelligence*, vol. 7, no. 5, pp. 1,203-1,225, 1993.

Efficient Color Histogram Indexing for Quadratic Form Distance Functions

James Hafner, Harpreet S. Sawhney, Will Equitz, Myron Flickner, and Wayne Niblack

***Abstract*—In image retrieval based on color, the weighted distance between color histograms of two images, represented as a quadratic form, may be defined as a match measure. However, this distance measure is computationally expensive (naively $O(N^2)$ and at best $O(N)$ in the number N of histogram bins) and it operates on high dimensional features ($O(N)$). We propose the use of low-dimensional, simple to compute distance measures between the color distributions, and show that these are lower bounds on the histogram distance measure. Results on color histogram matching in large image databases show that prefiltering with the simpler distance measures leads to significantly less time complexity because the quadratic histogram distance is now computed on a smaller set of images. The low-dimensional distance measure can also be used for indexing into the database.**

***Index Terms*—Color histogram matching, image querying, image databases, efficient multidimensionalal feature matching, histogram indexing.**

I. INTRODUCTION

In the *query by image content* (or QBIC) project, we are developing a system for efficient indexing and retrieval of images from a large database based on their content defined in terms of shapes, colors, textures, and user sketches [16]. Other efforts towards similar goals are presented in [2], [7], [8], [11], [12], [19].

Manuscript received May 5, 1993; revised Mar. 6, 1995.

J. Hafner, H.S. Sawhney, M. Flickner, and W. Niblack are with the Machine Vision Group, IBM Almaden Research Center, K54, 650 Harry Road, San Jose, CA 95120; e-mail sawhney@almaden.ibm.com.

W. Equitz is with Storm Technology, 1861 Landings Dr., Mountain View, CA 94043. This work was performed while this author was at the IBM Almaden Research Center.

IEEECS Log Number P95069.

In this paper we present efficient methods for retrieving images based on their color content. Color histograms are a way to represent the distribution of colors in images where each histogram bin represents a color in a suitable color space (RGB, Lab, etc.; see, for example, [17]). A distance or more precisely a pseudometric or pseudonorm, usually represented by a quadratic form, between a query image histogram and a data image histogram can be used to define the similarity match between the two distributions. However, because the histogram is typically a high-dimensional distribution ($N = 256$ or 64 colors, for instance), the distance measure is computationally expensive. A naive implementation of this as a quadratic form requires $O(N^2)$ operations, though this can be improved to $O(N)$ by some precomputations (e.g., by diagonalizing the quadratic form). Also, indexing on such high-dimensional features is typically not feasible. Moreover for large image databases, it is generally not feasible to compute the match measure against every image ($O(M)$ computation if M is the size of the database). One has to generate low-dimensional indices so that using standard database indexing methods (see for example [1]) retrieved involves only $O(\log M)$ comparisons. This is referred to as the *dimensionality curse* in multidimensionalal indexing in large databases [1]. Even efficient data structures for database indexing, like R^*-trees [18], work well for only up to 20 dimensions. Furthermore, the choice of low-dimensional features should satisfy the *completeness* [1] property. That is, the features should not lead to any false misses. Efficiency should not be at the expense of correctness.

We propose the use of low-dimensional, simple to compute distance measures between the color distributions, and show that these are *lower bounds* on the histogram distance measure in certain fairly general cases. Thus, similarity retrieval based on the cheaper measure achieves both the goals of low-dimensionality and completeness. Our results on color histogram matching in large image databases show that prefiltering with simpler distance measures leads to a considerable time saving because the quadratic form is now computed on a smaller set of images. Furthermore, the methodology is applicable to many other situations which involve computation of a quadratic form distance measure between two distributions.

II. The Problem

Let $\mathbf{x}$ and $\mathbf{y}$ be two N-dimensional distributions, color histograms for instance. For retrieval based on similarity of the two distributions, a distance (defined here to be a pseudometric or pseudonorm) between the two can be defined as a match measure. A weighted form of such a distance measure can be represented as a quadratic form:

$$d^2_{\text{hist}}(\mathbf{x}, \mathbf{y}) = (\mathbf{x}-\mathbf{y})^t \mathbf{A} (\mathbf{x}-\mathbf{y}), \tag{1}$$

where $\mathbf{A} = [a_{ij}]$ is a matrix and the weights a_{ij} denote similarity between bins i and j. These weights can be normalized so that $0 \le a_{ij} \le 1$, with $a_{ii} = 1$, and large a_{ij} denoting similarity between bins i and j, and small a_{ij} denoting dissimilarity. The two distributions can also be normalized so that $0 \le x_i, y_i \le 1$ and $\sum_i x_i = \sum_i y_i = 1$. It is to be emphasized here that d^2_{hist} can be a positive semidefinite form even when $\mathbf{A}$ is an indefinite matrix because of the normalization constraints on the distributions. For now it is assumed that the above equations indeed define a positive semidefinite form. It will be shown later that under certain conditions on the a_{ij}s, it always has this property. Besides, in any specific application or choice of matrix $\mathbf{A}$, this can be easily tested numerically

If the database records are images, and the histograms are the color distributions in the images, then d_{hist} represents a generalized quadratic histogram distance measure for color matching. Each bin or dimension of the histogram corresponds to a particular color in some chosen color space. Typically, 256 colors are adequate to capture the color distributions of most natural scenes. In contrast with direct Euclidean distance, the general quadratic form allows for similarity matching between different colors (represented by the histogram bins). (See the example in the next section.) Each entry a_{ij} in the "similarity matrix" $\mathbf{A}$ attempts to capture the perceptual similarity between the colors represented by bins i and j. This method of comparing histograms is more sophisticated than current popular methods such as that of Swain and Ballard [20], and, based on our experience with the QBIC system, more closely corresponds to human judgment of color similarity.

However, as has already been noted, the histogram quadratic form measure is computationally intensive and it operates on high dimensional features. The problem at hand is to define a considerably less expensive measure on considerably lower dimensional features so that:

1) the cheaper distance measure can be used to filter a large fraction of the database without any misses,
2) the expensive match measure computation can be limited to the small set of images retrieved, and
3) the database can be organized in terms of the low dimensional indices.

We show that, under fairly general conditions, the cheaper distance, call it d_k for now, can be bounded by d_{hist} in the form $\lambda d_k \le d_{\text{hist}}$ for some positive constant λ. Thus, in order to retrieve images satisfying $d_{\text{hist}} \le \varepsilon$, the inequality $d_k \le \varepsilon/\lambda$ can be used to retrieve images quickly and without misses. The expensive measure d_{hist} will then have to be applied only to the filtered set of images.

The application domain is discrete distribution matching, where we use additional information about the definitions of the distribution bins. Specifically, the similarity matrix A is used to construct a cheaper distance measure.

If the similarity information is ignored, then the problem is a straightforward distribution matching problem. Swain and Ballard [20] treat their color matching problem in this way, and use L_1 distance as a measure of the distance between two histograms. Other obvious candidates are the "relative entropy" or "Kullback-Leibler" distance [3] or the distance implied by the standard chi-squared statistical test [15]. Ioka [10] used the histogram quadratic form distance, but did not use the cheaper distance as a lower bound to efficiently filter out unwanted records.

III. Histogram Distance

Given the N-dimensional normalized distributions $\mathbf{x}$ and $\mathbf{y}$, if $\mathbf{z} = \mathbf{x} - \mathbf{y}$, then $-1 \le z_i \le 1$, $\sum_i z_i = 0$, and $d^2_{\text{hist}}(\mathbf{x}, \mathbf{y}) = \mathbf{z}^T \mathbf{A} \mathbf{z}$. We call the set of such $\mathbf{z}$s the histogram space (though it is not technically a linear space).

Without loss of generality, $\mathbf{A}$ can be assumed to be a symmetric matrix because the antisymmetric part does not contribute to the quadratic form.

An example will illustrate why similarity weighted comparison between color histograms leads to perceptually desirable results in contrast with comparison based on direct Euclidean distance between the distributions. For simplicity, consider a histogram distribution of three colors, say red, orange, and blue, with

$$\mathbf{A}_{\text{red, orange, blue}} = \begin{bmatrix} 1.0 & 0.9 & 0.0 \\ 0.9 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

where red and orange are considered highly similar. Consider a pure red image, $\mathbf{x} = [1.0, 0.0, 0.0]^T$, and a pure orange image, $\mathbf{y} = [0.0, 1.0, 0.0]^T$. The (squared) histogram distance of (1) is 0.2. This low distance reflects the perceptual similarity of the two images although their distributions populate distinct bins of the histogram so that their squared Euclidean distance is 2.0.

It is now shown that for certain choices of A, d^2_{hist} is indeed nonnegative on the histogram space.

First note that a sufficient condition for d^2_{hist} to be nonnegative is that the matrix A be positive semidefinite (PSD). But this assumption is not *necessary* because of the condition $\sum_i z_i = 0$. In fact, in our experiments the As chosen are not PSD (or PD).

Next, it is shown in [5] that a quadratic form $\mathbf{z}^T\mathbf{H}\mathbf{z}$, $\mathbf{H} = [h_{ij}]$, on the subspace $\sum_i z_i = 0$ is negative semidefinite ($\mathbf{z}^T\mathbf{H}\mathbf{z} \le 0$) *if* each h_{ij} represents the distance between some points P_i and P_j in some finite dimensional L_1 or L_2 normed space. In particular, we have 1) $h_{ii} = 0$, 2) $h_{ij} = h_{ji}$, and 3) $h_{ij} \le h_{ik} + h_{kj}$. This holds for arbitrary N (the dimensionality of $\mathbf{H}$) and is also independent of the dimensionality of the space over which the points P_i are defined.

Now, let d_{ij} be the Euclidean (L_2) distance between colors i and j in some color space, for instance, [R(ed), G(reen), B(lue)], or [L, u, v] or the Munsell color space. Therefore d_{ij} satisfy the requirements (stated above) for h_{ij}.

We show that for one reasonable choice of a_{ij} in terms of d_{ij}, d^2_{hist} indeed is nonnegative. Let $d_{\max} = \max_{i,j}(d_{ij})$ and

$$a_{ij} = (1 - d_{ij}/d_{\max}). \tag{2}$$

Then $d^2_{\text{hist}} = \mathbf{z}^T\mathbf{A}\mathbf{z} = \sum_{i,j} z_i z_j (1 - d_{ij}/d_{\max})$. Given that $\sum_i z_i = 0$, we have $d^2_{\text{hist}} = -\sum_{ij} z_i z_j d_{ij}/d_{\max} = -(1/d_{\max})\mathbf{z}^T\mathbf{H}\mathbf{z}$, where $\mathbf{H} = (d_{ij})$. Using the result stated above, $\mathbf{z}^T\mathbf{H}\mathbf{z} \le 0$ so that $d^2_{\text{hist}} \ge 0$.

Alternatively, another choice for a_{ij} is

$$a_{ij} = \exp(-\sigma(d_{ij}/d_{\max})^2) \tag{3}$$

for some positive constant σ. These a_{ij}s enforce a faster roll-off as a function of d_{ij}. Clearly, for σ sufficiently large, each of these matrices is PD since it becomes diagonally dominant. For the specific values d_{ij} and σ we have used in practice, these matrices have not always been PD, but have had the property that the induced quadratic form d^2_{hist} is nonnegative. This was verified numerically in our examples but is probably true in general though we do not have a proof of this.

Other choices of similarity metrics that capture perceptual similarity of colors and make $d^2_{\text{hist}} \ge 0$ are valid also. We have used the forms (2) and (3) in our experiments on real images.

Now, d^2_{hist} can be written in a different form by eliminating the constraint $\sum_i z_i = 0$

$$d^2_{\text{hist}} = \mathbf{z}^T\mathbf{A}\mathbf{z} = \left[\tilde{\mathbf{z}}^T z_N\right]\begin{bmatrix} \mathbf{A}_{N-1} & \mathbf{a}_{*N} \\ \mathbf{a}^T_{*N} & a_{NN} \end{bmatrix}\begin{bmatrix} \tilde{\mathbf{z}} \\ z_N \end{bmatrix},$$

where $\tilde{\mathbf{z}}^T = [z_1 \cdots z_{N-1}]$ and $\mathbf{A}$ has been decomposed into its top left $(N-1) \times (N-1)$ component and the rest ($\mathbf{a}_{*N}$ being the Nth column of A less the last entry a_{NN}). Applying $\sum_i z_i = 0$ to this, we get

$$d^2_{\text{hist}} = \tilde{\mathbf{z}}^T\left[\mathbf{A}_{N-1} - \mathbf{a}_{*N}\cdot\mathbf{1}^T - \mathbf{1}\cdot\mathbf{a}^T_{*N} + a_{NN}\mathbf{1}\cdot\mathbf{1}^T\right]\tilde{\mathbf{z}} \tag{4}$$

which is written as $d^2_{\text{hist}} = \mathbf{z}^T\mathbf{A}\mathbf{z} = \tilde{\mathbf{z}}^T\tilde{\mathbf{A}}\tilde{\mathbf{z}}$, where

$$\tilde{\mathbf{A}} = \left[\mathbf{A}_{N-1} - \mathbf{a}_{*N}\cdot\mathbf{1}^T - \mathbf{1}\cdot\mathbf{a}^T_{*N} + a_{NN}\mathbf{1}\cdot\mathbf{1}^T\right] \tag{5}$$

is an $(N-1) \times (N-1)$ modified similarity matrix, and $\mathbf{1}$ is a vector of $N-1$ ones.

The only constraint on $\tilde{\mathbf{z}}$ is that it lie within the polytope defined by $\sum_i |\tilde{z}_i| \le 1$. (For instance, in two dimensions, this defines the unit "diamond" polygon.) Thus, now only a constraint on the L_1-norm of zs is left. Therefore, with the results on the positiveness of d^2_{hist} shown earlier, (4) implies that

1) $d^2_{\text{hist}} \ge 0$ *if and only if* $\tilde{\mathbf{A}}$ is PSD, and
2) for the choice of a_{ij} in (2), $\tilde{\mathbf{A}}$ *is* PSD.

As already mentioned, the choice in (3) resulted in the corresponding $\tilde{\mathbf{A}}$ being PSD, in all our examples.

We will call histograms $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ (respectively, the corresponding vector $\tilde{\mathbf{z}}$) *normalized* because $\sum_i \tilde{x}_i \le 1$ (respectively, $\sum_i |\tilde{z}_i| \le 1$), and *reduced* since they are dimension $N-1$.

IV. Average Color Distance

In this section, we present a particularly intuitive distance measure, called the average color distance (the distance between average colors), as a first instance of a cheaper measure that satisfies the requirements outlined in Section II. Subsequently, this distance measure will be generalized to a series of low-dimensional distance measures, each presenting a trade-off between their dimensionality and the number of false hits.

Given that each bin of a color histogram represents a three-dimensional color vector in a suitably defined color space, the average color of an image histogram is defined to be the weighted average color corresponding to the normalized color histogram distribution. Specifically, let $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2 \cdots \mathbf{c}_N]$ be a $3 \times N$ matrix whose ith column is the color $\mathbf{c}_i = [\alpha_i\ \beta_i\ \gamma_i]^T$, where α, β, and γ represent the magnitudes along the three color dimensions (R, G, B, for instance). Given two N-dimensional color histograms, $\mathbf{x}$ and $\mathbf{y}$, the 3×1 average color vector for each is

$$\mathbf{x}_{avg} = \mathbf{C}\mathbf{x}, \quad \mathbf{y}_{avg} = \mathbf{C}\mathbf{y}.$$

The squared average color distance is defined by

$$d^2_{\text{avg}} = \left(\mathbf{x}_{avg} - \mathbf{y}_{avg}\right)^T\left(\mathbf{x}_{avg} - \mathbf{y}_{avg}\right) = \mathbf{z}^T\mathbf{C}^T\mathbf{C}\mathbf{z}.$$

With $\mathbf{W} = \mathbf{C}^T\mathbf{C}$, $d^2_{\text{avg}} = \mathbf{z}^T\mathbf{W}\mathbf{z} = \tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}}$, where $\tilde{\mathbf{z}}$ is as defined in Section III and $\tilde{\mathbf{W}}$ is defined in terms of $\mathbf{W}$ similar to $\tilde{\mathbf{A}}$ of (5). Clearly by construction $\mathbf{W}$ and $\tilde{\mathbf{W}}$ are PSD (but *not* PD since their rank is at most three).

Note that d_{avg} is defined over 3-dimensional features, $\mathbf{x}_{avg}$ and $\mathbf{y}_{avg}$, in contrast with N-dimensional (typically 256 or 64) features for d_{hist}. Also, the average color can be precomputed (at database population/compilation time) and then it can be organized into an indexable (R^*-tree like) structure. Furthermore, d_{hist} can be bounded from below by a simple function of d_{avg} which ensures that indexing on d_{avg} will be without any false misses. We show this in the next section.

A. Bound Between d_{hist} and d_{avg}

If $\tilde{\mathbf{A}}$ is PD, then d_{hist} is just a norm on the histogram space, which is a subset of a linear space. The matrix **C** maps this vector space into Euclidean 3-space with the standard norm. Thus, d_{avg} is just the Euclidean norm on the image of the histogram space under the linear map induced by **C**. In this context, the induced linear map is a bounded linear operator. Consequently,

$$d_{avg} \le \|\mathbf{C}\| d_{hist},$$

where $\|\mathbf{C}\|$ is the norm of this linear operator. In other words, the existence of the required inequality is a consequence of a general theorem [6, p. 57] about finite dimensional normed vector spaces and bounded linear operators. If, as in our case, neither $\tilde{\mathbf{A}}$ nor $\tilde{\mathbf{W}}$ are PD but only PSD then the above results do not apply directly, though they can be extended to this case by restricting to the subspace on which $\tilde{\mathbf{A}}$ is PD. However, this does not provide us with a constructive method for computing the multiplier. In the following, we give an alternate proof extending the results to this case and including a simple method for computing the multiplier using standard results in constrained optimization.

THEOREM 1. *With d_{hist} and d_{avg} defined as above, if $\tilde{\mathbf{A}}$ is positive semidefinite, then for all vectors* **x** *and* **y**, $d_{hist}^2 \ge \lambda_1 d_{avg}^2$, *where λ_1 is the minimum eigenvalue of the generalized eigenvalue problem* $\tilde{\mathbf{A}}\tilde{\mathbf{z}} = \lambda\tilde{\mathbf{W}}\tilde{\mathbf{z}}$.

PROOF. It will be shown that $\tilde{\mathbf{z}}^T\tilde{\mathbf{A}}\tilde{\mathbf{z}} \ge \lambda_1\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}}$. Let $\eta > 0$. Because $\tilde{\mathbf{A}}$ is PSD there is a unique solution to the constrained minimization problem

$$\min_{\tilde{\mathbf{z}}:\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}}=\eta} \tilde{\mathbf{z}}^T\tilde{\mathbf{A}}\tilde{\mathbf{z}}. \tag{6}$$

This solution occurs at a value of $\tilde{\mathbf{z}}$ where the function $\tilde{\mathbf{z}}^T\tilde{\mathbf{A}}\tilde{\mathbf{z}} - \lambda(\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}} - \eta)$ has a critical value [13, Section 10.3]. These critical values (in the variables $\tilde{\mathbf{z}}$ and λ) occur when $\tilde{\mathbf{A}}\tilde{\mathbf{z}} = \lambda\tilde{\mathbf{W}}\tilde{\mathbf{z}}$ and $\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}} = \eta$, and these are regular points because $\eta > 0$. Let $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_{N-1}$, be the generalized eigenvalue solutions of the first equation and $\tilde{\mathbf{z}}_i$ be any generalized eigenvector also satisfying the second condition. Because the minimum in (6) exists, it must be the case that for some i, $1 \le i \le N-1$, $\tilde{\mathbf{z}}^T\tilde{\mathbf{A}}\tilde{\mathbf{z}} \ge \tilde{\mathbf{z}}_i^T\tilde{\mathbf{A}}\tilde{\mathbf{z}}_i$, for all $\tilde{\mathbf{z}}$ such that $\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}} = \eta$. But the right hand side of this equation is just $\lambda_i\eta$. Hence this minimum occurs for $i = 1$, the minimum eigenvalue.

We have shown that

$$\tilde{\mathbf{z}}^T\tilde{\mathbf{A}}\tilde{\mathbf{z}} \ge \lambda_1\eta = \lambda_1\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}},$$

for $\tilde{\mathbf{z}}$ such that $\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}} = \eta$, with $\eta > 0$ being completely arbitrary. If $\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}} = 0$, then this inequality holds a priori because $\tilde{\mathbf{A}}$ is PSD. Consequently, $d_{hist}^2 \ge \lambda_1 d_{avg}^2$, as claimed.

Note that λ_1 does not depend on the data, that is the image histograms, but only on the similarity matrix **A** and on the definitions of the colors in the matrix **C**. In our experiments, we used the IMSL routine [9, p. 454], *DGVLRG*, to compute λ_1. This routine does not require $\tilde{\mathbf{W}}$ or $\tilde{\mathbf{A}}$ to be positive definite.

Because both $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{W}}$ are PSD, $\lambda_1 \ge 0$. It is indeed possible that $\lambda_1 = 0$, even though this is of no use to us in the application. This will occur when the null space of $\tilde{\mathbf{A}}$ is not contained in the null space of $\tilde{\mathbf{W}}$. This case is ignored throughout the rest of the paper. Furthermore, for $\lambda_1 > 0$, **C** (and so $\tilde{\mathbf{W}}$) could be normalized so that $\lambda_1 = 1$.

As mentioned before, since $d_{hist}^2 \ge \lambda_1 d_{avg}^2$, for any *range query* of the form $d_{hist} \le \varepsilon$, images retrieved using the filter $d_{avg} \le \varepsilon/\sqrt{\lambda_1}$ are guaranteed to be a super set of the complete target set. If the filtering returns a relatively small fraction of the entire database, then the expensive d_{hist} needs to be computed only for that small fraction and not for the whole database.

Note that if the 3-dimensional average color for each image, (**Cx**), is precomputed and stored, then at query time the average color distance requires only three multiplications and subtractions, and two additions, i.e., is independent of N, the number of histogram bins.

V. EXPERIMENTAL RESULTS WITH d_{avg}

The quadratic distance, d_{hist}, has been in use in the QBIC system [16] for retrieval of images based on similarity of color histograms. Experiments with a variety of color spaces found that the best performance (using human judgment of the similarity between the query and database images) was obtained using a variant of the Munsell color space. This is a 3-dimensional space with the property that the Euclidean distance between two color vectors corresponds over a large part of the space to the perceptual difference between the colors. A transformation from the (R, G, B) space, in which the original images are defined, to the Munsell space, described in [14], was used. In the experiments reported here, 256 dimensional histograms were used.

We performed simulations to evaluate the effectiveness of filtering with d_{avg} on a database of 917 assorted natural images in the QBIC database. The relative retrieval efficiency between two methods is reported: 1) simple sequential evaluation of d_{hist} for all database vectors, (referred to as '*naive*'), and 2) filtering using d_{avg} followed by evaluation of d_{hist} only on those records that pass through the filter (referred to as '*filtered*'). Indexing methods (such as R^*-trees) were not used because the focus was on the gains from the filtering step. The sample queries involved matching each histogram record against the remaining records.

In the ideal case, the filtering step would filter out only exactly those records for which d_{hist} was in the desired range. It is guaranteed that filtering will include all records for which this is true, but some "false alarms" will also be retrieved. The color similarity matrix **A** was defined with a_{ij} as in (2). Thus, **A** is guaranteed to be nonnegative definite.

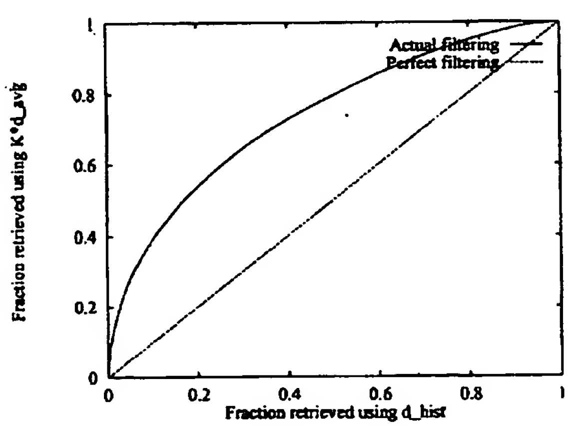


Fig. 1. d_{hist} vs. d_{avg} retrieval.

Fig. 1 shows the extent of the false alarms generated by d_{avg} for the test database. For instance, if the tolerance is set so that the top 10% of the database would have been retrieved by direct search using d_{hist}, then the filtering with d_{avg} retrieved approximately 40% of the database. Even this filtering results in a considerable saving because d_{hist} is applied only to the filtered 40% of the records and not to the complete 100%.

VI. Generalizing Cheap Similarity Measures

It should be evident from the discussion in Section IV that d_{avg} is not the only distance that leads to low-dimensionality features and filtering of the database without any misses. It just happens to be one that results in an intuitively appealing feature, the average color. No particular assumptions were made on the matrix **C**. Therefore, the question to ask is if the notion of a cheaper distance measure can be generalized.

In this section, we pose the question that given a particular number, k of feature dimensions, what is the optimum k-dimensional index and the corresponding distance? A requirement is that the distance measure should be based on the *precomputed* k-dimensional features like the average color. Similar to the definition of average color, let $\mathbf{x}_k = \mathbf{U}_k\tilde{\mathbf{z}}$ be a k-dimensional feature defined for the $(N-1)$-dimensional normalized and reduced histogram $\tilde{\mathbf{z}}$, where $\mathbf{U}_k$ is a $k \times (N-1)$ matrix. The k-distance (like d_{avg}) between two histograms $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ is determined by:

$$d_k^2 = d_k^2(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = (\mathbf{x}_k - \mathbf{y}_k)^T(\mathbf{x}_k - \mathbf{y}_k) = \tilde{\mathbf{z}}^T\mathbf{U}_k^T\mathbf{U}_k\tilde{\mathbf{z}} = \tilde{\mathbf{z}}^T\tilde{\mathbf{A}}_k\tilde{\mathbf{z}},$$

where $\tilde{\mathbf{A}}_k = \mathbf{U}_k^T\mathbf{U}_k$ is an at most k-rank PSD matrix.

The problem of determining the optimal d_k for a fixed k can be formalized as finding the rank k matrix $\mathbf{A}_k$ at which the extremum

$$\inf_{\mathbf{A}_k}\sup_{\tilde{\mathbf{z}}} \tilde{\mathbf{z}}^T(\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k)\tilde{\mathbf{z}} \tag{7}$$

is attained, subject to

1) $(\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k)$ being PSD, and
2) $\sum_i |\tilde{z}_i| \le 1$.

In words, we want to find that k-rank approximation to $\tilde{\mathbf{A}}$ such that the maximum difference (over $\tilde{\mathbf{z}}$) between the true distance and the lower dimensional distance is a minimum for the given k. Also, because we want $d_{hist} \ge d_k$ to avoid false misses (with $\lambda = 1$), $(\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k)$ should be PSD.

The Singular Value Decomposition (SVD) (see, for example, [4]) of $\tilde{\mathbf{A}}$ provides a constructive answer to the above problem. Let $\tilde{\mathbf{A}} = \mathbf{V}^T\mathbf{\Sigma}\mathbf{V}$ be the SVD of $\tilde{\mathbf{A}}$, where $\mathbf{V}$ is an orthogonal matrix and $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_{N-1})$ with $\sigma_1 \ge \cdots \ge \sigma_k \ge \cdots \ge \sigma_{N-1} \ge 0$. Then, a solution to the problem in (7) is given by

$$\tilde{\mathbf{A}}_k = \mathbf{V}_k^T\mathbf{\Sigma}_k\mathbf{V}_k \tag{8}$$

where $\mathbf{\Sigma}_k$ is the $k \times k$ diagonal matrix $\mathrm{diag}(\sigma_1, \ldots, \sigma_k)$, and $\mathbf{V}_k$ is the matrix whose rows are the first k rows of $\mathbf{V}$, i.e., corresponding to the first k singular values.

To see this, suppose for a moment that the singular values σ_j, $j = 1, \ldots, N-1$ are *not* ordered. The results in [4] show that an optimal k-rank approximation $\tilde{\mathbf{A}}_k$ to a matrix $\tilde{\mathbf{A}}$ must have the property that $\mathbf{V}\tilde{\mathbf{A}}_k\mathbf{V}^T$ is diagonal and have entries which come from the diagonal matrix $\mathbf{\Sigma}$, i.e., the singular values of $\tilde{\mathbf{A}}$. Consequently, the extremum in (7) can only be attained if $\tilde{\mathbf{A}}_k$ has the form given in equation (8). Now observe that under this assumption

$$d_{diff} = \tilde{\mathbf{z}}^T(\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k)\tilde{\mathbf{z}} = (\mathbf{V}\tilde{\mathbf{z}})^T(\mathbf{\Sigma} - \mathbf{\Sigma}_k')(\mathbf{V}\tilde{\mathbf{z}}),$$

where

$$\mathbf{\Sigma}_k' = \begin{pmatrix} \mathbf{\Sigma}_k & 0 \\ 0 & 0 \end{pmatrix}.$$

From this we easily see that $(\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k)$ is PSD. Furthermore, the level set $d_{diff} = 1$ defines a hyperellipsoid in $N - 1 - k$ dimensions whose principal axes are the square-roots of the reciprocals of $\sigma_{k+1}, \ldots, \sigma_{N-1}$ (without loss of generality, we assume that these sigmas are non-zero else the subspace can be further reduced in dimension). The constraint $\sum_i |\tilde{z}_i| \le 1$ defines a polytope with corners on each of the $\tilde{z}_i$ axes. The maximum of d_{diff} over such $\tilde{\mathbf{z}}$ will lie on one of the corners. The value of the maximum will be determined by the smallest principal axis of the ellipsoid, namely $\min_{k+1 \le j \le N-1}\{1/\sqrt{\sigma_j}\}$. The minimum, over all possible orderings of the σ's, of this maximum occurs when the smallest axis is largest. That is, the desired extremum occurs when $\sigma_1 \ge \cdots \ge \sigma_k \ge \cdots \ge \sigma_{N-1}$, as claimed above. Thus, $\tilde{\mathbf{A}}_k$ of (8) is indeed the solution to (7).

A. k-dimensional Index and d_k

The above result creates a method for constructing a k-dimensional index or feature (analogous to the 3-dimensional average color features) for similarity search on N-dimensional histograms. First, off-line the SVD decomposition of the similarity matrix is calculated and a small integer k is chosen. (For best results, the choice of k should depend on the distribution of the significant singular values of the similarity matrix.) This determines the matrices V, Σ and V_k and Σ_k. During database population time, an image is processed and the following data is collected: 1) the color histogram with N buckets, 2) the N-dimensional feature vector obtained by multiplying the histogram by V, 3) and the k-dimensional feature vector obtained by multiplying by V_k (more precisely by truncating the N dimensional feature vector). The k-dimensional features are organized in the database for search matching on this low dimensional feature and the N-dimensional features are stored along with them.

At matching time, the query histogram specification is converted to both the N and k-dimensional features just as above. For a range query of the type we are dealing with, matching is done first on the k-dimensional features. This requires at most $O(k)$ steps per image or database record. If a database record passes the filter step on this k-dimensional matching, then and only then is the full matching done on the N dimensional data.

There are two points here. First, we require at most $O(k)$ operations per database record to perform the filter step. This is then followed by the $O(N)$ operations to perform the final matching, but only on the small set of filtered images. Second, by structuring the database on the k-dimensional features in an efficient manner (R^*-tree, for example) the k-dimensional matching need not be performed on every element in the database. This extra efficiency is not possible on high dimensional features.

Furthermore, in a large database, the cost of retrieving large records can overshadow even the cost of the full distance computation. On the other hand, for low dimensional searches, even sequential search accesses small records. Consequently, this filtering method can reduce computation costs both in disk access and distance computation. The only assumption here is that $\tilde{\mathbf{A}}$ has only a few significant singular values, so that the number of false alarms will be relatively small. As is shown in the next section, this indeed is the case in the $\tilde{\mathbf{A}}$ s that we use in our experiments.

VII. Experimental Results with d_k

In this section we present experimental results for the selectivity of the filtering step using d_k for various values of k. The range of singular values of the matrix $\tilde{\mathbf{A}}$ which we used is 150.78 to 0.03; its condition number is 5,026. The trend of the variation in singular values is shown as a plot in Fig. 2. The first 12 singular values (in descending order) of the 255 × 255 matrix range from 150.78 to 1.01. All other singular values are less than 1.0.

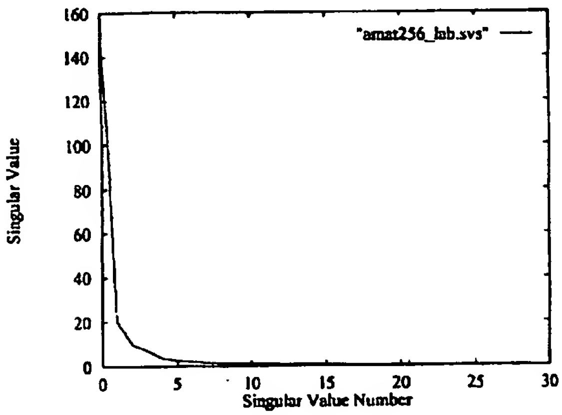


Fig. 2. First few significant singular values of $\tilde{\mathbf{A}}$.

Figs. 3–5 show the retrieval efficiency achieved using the largest 3, 9, and 15 singular values, respectively, to synthesize $\tilde{\mathbf{A}}_k$. The figures clearly show the rapid gain in efficiency of filtering with d_k as the dimension of the k-index increases. For instance, for a target retrieval of 10% of the images using d_{hist}, the fraction of images retrieved using d_k is about 40% for $k = 3$, 18% for $k = 9$, and 15% for $k = 15$. The efficiency with $k = 3$ is almost identical to that obtained with d_{avg} (there is probably a reason for this but we have not investigated it in depth). But with a small increase in the dimensionality to 9, the number of false retrievals falls from 30% to 8%. The marginal efficiency increase tapers off with further increase in k.

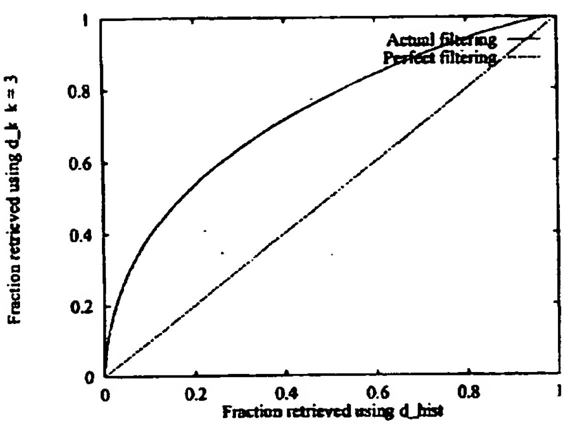


Fig. 3 d_{hist} vs. d_k retrieval for $k = 3$

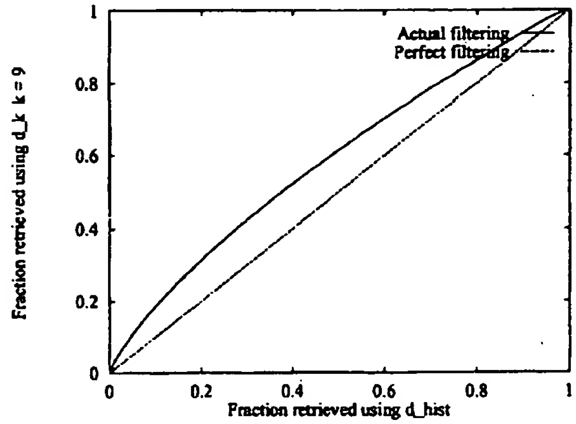


Fig. 4. d_{hist} vs. d_k retrieval for $k = 9$.

Fig. 6 shows the ratio of retrieval using d_k to retrieval using d_{hist}, for 10% retrieval with d_{hist}, as a function of k. The asymptotic behavior of the ratio (efficiency) is evident from this figure.

In order to illustrate the efficiency gained in CPU time only, Fig. 7 shows the relative CPU times (time taken for distance computation only), for up to 10% retrieval, between d_{hist} computation over the entire database, and d_{hist} computation over only the part of the database filtered using d_{avg}. Similarly, Fig. 8, shows the CPU times for filtering with d_k with $k = 15$. The CPU time for a 256-bin d_{hist} was about 1*ms*. on an RS6000/350. It is clear from the comparisons that even for retrievals as high as 10% of the database, d_{avg} filtering

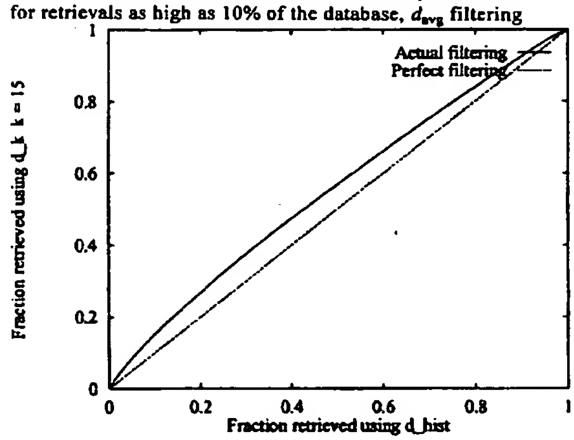


Fig. 5. d_{hist} vs. d_k retrieval for $k = 15$.

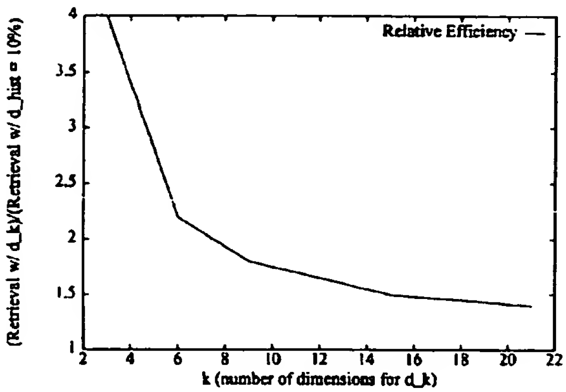


Fig. 6. Retrieval ratio with d_k to d_{hist} vs. k (10% of database retrieved by d_{hist}).

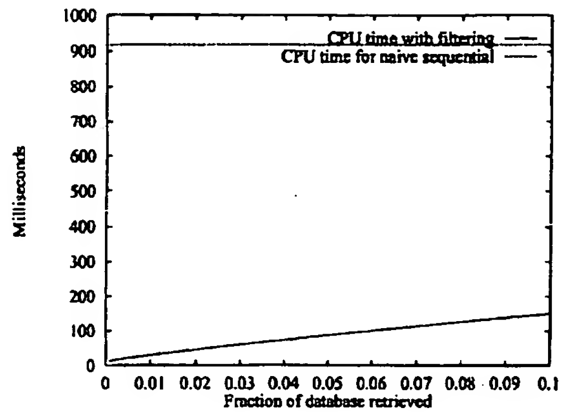


Fig. 8. CPU time for *naive* retrieval vs. filtered retrieval with d_k for $k = 15$.

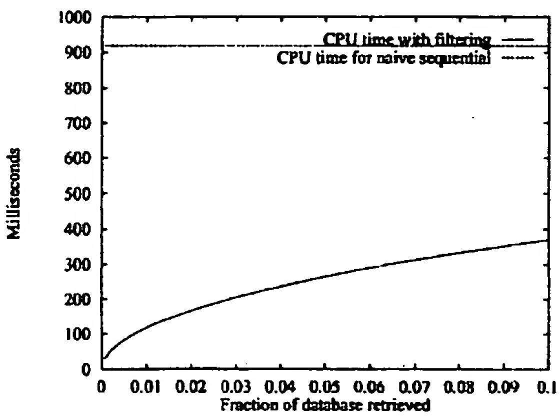


Fig. 7. CPU time for *naive* retrieval vs. filtered retrieval with d_{avg}.

results in almost 1/3rd the CPU time, and d_k ($k = 15$) results in about 1/6th the CPU time of naive d_{hist} retrieval. It is assumed in this comparison that both for direct d_{hist} querying and filtered querying, all the database records are read into memory before the distance computations. Given that the focus of this paper is on the filtering efficiency, a detailed timing comparison with realistic models and simulations of efficient indexing, combined disk access and CPU times has been left for another time.

VIII. Conclusions

We presented a method for efficient retrieval of images based on similarity of color histograms. Clearly, the method may also be useful for any other application involving a quadratic distance measure. The results indicate that the simple filtering method described here easily outperforms the naive histogram comparison method. Using low-dimensional indexing techniques (e.g., k-index) would certainly make our method more attractive for very large database applications.

In the future, we plan to understand analytically the relationship between the ratio of retrievals using d_k to that using d_{hist}. Such an analysis would require assumptions on the distributions of the histograms themselves, for instance, that they are drawn from a uniform distribution. We would also like to verify our conjecture that d_{hist} is indeed nonnegative for the choice of a_{ij} in (3). Finally, we will build an indexable color image database using the k-index and demonstrate the extra efficiency in retrieval times.

References

[1] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," *Fourth Int'l Conf. Foundations of Data Organization and Algorithms*, Chicago, Ill., 1993.

[2] E. Binaghi, I. Gagliardi, and R. Schettini, "Indexing and fuzzy logic-based retrieval of color images," *Visual Database Systems, II, IFIP Trans. A-7*, pp. 79-92, Elsevier Science Publishers, 1992.

[3] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, 1991.

[4] P. Dewilde and E.F. Deprettere, "Singular value decomposition, an introduction," *SVD and Signal Processing: Algorithms, Applications, and Architectures*, E.F. Deprettere, ed., pp. 3-42. New York: Elsevier Science Publishing Co., 1988.

[5] T. Feder, "On a quadratic form for points in space," IBM Internal Note, 1993.

[6] G.H. Golub and C.F. van Loan, *Matrix Computations*. The Johns Hopkins Univ. Press, 1989.

[7] J. Hasegawa, N. Okada, and J. Toriwaki, "Intelligent retrieval of chest x-ray database using sketches," *Systems and Computers in Japan*, vol. 20, no. 7, pp. 29-42, 1989.

[8] K. Hirata and T. Kato. "Query by visual example," *Adv. in Database Technology EDBT '92, Third Int'l Conf. on Extending Database Technology*. Vienna, Austria: Springer-Verlag, Mar. 1992.

[9] IMSL, Inc., *User's Manual MATH/Library*. Texas: IMSL, Inc., 1991.

[10] M. Ioka, "A method of defining the similarity of images on the basis of color information," Technical report RT-0030, IBM Tokyo Research Lab, 1989.

[11] M.A. Ireton and C.S. Xydeas, "Classification of shape for content retrieval of images in a multimedia database," *Sixth Int'l Conf. on Dig. Proc. of Signals in Comm.*, pp. 111-116, Loughborough, U.K., Sept. 2-6, 1990.

[12] H.V. Jagadish, "A retrieval technique for similar shapes," *Int'l Conf. on Management of Data, SIGMOD 91*, pp. 208-217, Denver, Colo., May 1991.

[13] D.G. Luenberger, *Linear and Nonlinear Optimization*. Mass.: Addison-Wesley, 1984.

[14] M. Miyahara and Y. Yoshida, "Mathematical transform of (R, G, B) color data to Munsell (H, V, C) color data," *Visual Communication and Image Processing*, vol. 1,001, pp. 650-657, SPIE, 1988.

[15] A. Mood, F. Graybill, and D. Boes, *Introduction to the Theory of Statistics*. McGraw-Hill series in probability and statistics, McGraw-Hill, third edition, 1963.

[16] W. Niblack, R. Berber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yanker, "The QBIC project: Querying ims. by content using color, texture, and shape," *SPIE 1908, Storage and Retrieval for Image and Video Dbases*, Feb. 1993.

[17] C.L. Novak and S.A. Shafer, "Color vision," *Physics-Based Vision COLOR*, pp. 1-10, Jones and Bartlett, 1992.

[18] M. Otterman, "Approximate matching with high dimensionality R-trees," MSc scholarly paper, Dept. of Computer Science, Univ. of Maryland, College Park, Md., supervised by C. Faloutsos, 1992.

[19] A. Pentland, R. Picard, G. Davenport, and B. Welsh, "The BT/MIT project on advanced image tools for telecommunications: An overview," Technical report 212, MIT Media Lab. Perceptual Comp. Group TR, 1992.

[20] M.J. Swain and D.H. Ballard, "Color indexing," *Int'l J. Computer Vision*, vol. 7, no. 1, pp. 11-32, 1991.

Efficient color histogram indexing

Sawhney, H.S. Hafner, J.L.
Machine Vision Group, IBM Almaden Res. Center, San Jose, CA, USA;

This paper appears in: **Image Processing, 1994. Proceedings. ICIP-94., I International Conference**

Meeting Date: 11/13/1994 - 11/16/1994
Publication Date: 13-16 Nov. 1994
Location: Austin, TX USA
On page(s): 66 - 70 vol.2
Volume: 2
Reference Cited: 14
Inspec Accession Number: 5010118

Abstract:
In image retrieval based on **color**, the weighted distance between **color histo** two images, represented as a quadratic form, may be defined as a match mea However, this distance measure is computationally expensive (naively O(N^2) a O(N) in the number N of **histogram** bins) and it operates on high dimensiona (O(N)). We propose the use of low-dimensional, simple to compute distance m between the **color** distributions, and show that these are lower bounds on the **histogram** distance measure. Results on **color histogram** matching in large databases show that pre-filtering with the simpler distance measures leads to significantly less time complexity because the quadratic **histogram** distance i computed on a smaller set of images. The low-dimensional distance measure used for indexing into the database

Index Terms:
color distributions **color histogram** indexing **color histogram** matching colour ima computational complexity database indexing database theory **histogram** distance m image colour analysis image matching indexing information retrieval large image d low-dimensional distance measures lower bounds match measure pre-filtering qua **histogram** distance time complexity visual databases weighted distance **color** dis **color histogram** indexing **color histogram** matching colour image retrieval compu complexity database indexing database theory **histogram** distance measure imag analysis image matching indexing information retrieval large image databases lo dimensional distance measures lower bounds match measure pre-filtering quadra

distance time complexity visual databases weighted distance

Documents that cite this document

Select link to view other documents in the database that cite this one.

Search Results [PDF FULL-TEXT 424 KB] NEXT DOWNLOAD CITATION

EFFICIENT COLOR HISTOGRAM INDEXING

Harpreet S. Sawhney and James L. Hafner

Machine Vision Group
IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120
Net: sawhney/hafner@almaden.ibm.com

Abstract

In image retrieval based on color, the weighted distance between color histograms of two images, represented as a quadratic form, may be defined as a match measure. However, this distance measure is computationally expensive (naively $O(N^2)$ and at best $O(N)$ in the number N of histogram bins) and it operates on high dimensional features ($O(N)$). We propose the use of low-dimensional, simple to compute distance measures between the color distributions, and show that these are lower bounds on the histogram distance measure. Results on color histogram matching in large image databases show that pre-filtering with the simpler distance measures leads to significantly less time complexity because the quadratic histogram distance is now computed on a smaller set of images. The low-dimensional distance measure can also be used for indexing into the database.

1. INTRODUCTION

In the *query by image content* (or QBIC) project, we are developing a system for efficient indexing and retrieval of images from a large database based on their content defined in terms of shapes, colors, textures and user sketches [12].

In this paper we present efficient methods for retrieving images based on their color content. A distance or more precisely a pseudo-metric or pseudo-norm, usually represented by a quadratic form, between a query image color histogram and a data image histogram can be used to define the similarity match between the two color distributions. However, because the histogram is typically a high-dimensional distribution ($N = 256$ or 64 colors, for instance), the distance measure is computationally expensive. A naive implementation of this as a quadratic form requires $O(N^2)$ operations, though this can be improved to $O(N)$ by some precomputations. Also, indexing on such high-dimensional features is typically not feasible. Moreover for large image databases, it is generally not feasible to compute the match measure against every image ($O(M)$ computation if M is the size of the database). One has to generate low-dimensional indices so that retrieval involves only $O(\log M)$ comparisons. Even efficient data structures for database indexing, like R^*-trees [13], work well for only up to 20 dimensions. Furthermore, the choice of low-dimensional features should satisfy the *completeness* [1] property. That is, the features should not lead to any false misses. Efficiency should not be at the expense of correctness.

We propose the use of low-dimensional, simple to compute distance measures between the color distributions, and show that these are *lower bounds* on the histogram distance measure in certain fairly general cases. Thus, similarity retrieval based on the cheaper measure achieves both the goals of low-dimensionality and completeness. This applies to any matching problem involving quadratic form distance measure between two distributions.

2. THE PROBLEM

Let $\mathbf{x}$ and $\mathbf{y}$ be two N-dimensional distributions, color histograms for instance. For retrieval based on similarity of the two distributions, a distance (defined here to be a pseudo-metric or pseudo-norm) between the two can be defined as a match measure. A weighted form of such a distance measure can be represented as a quadratic form:

$$d_{\text{hist}}(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^t \mathbf{A}(\mathbf{x}-\mathbf{y}), \tag{1}$$

where $\mathbf{A} = [a_{ij}]$ is a matrix and the weights a_{ij} ($0 \le a_{ij} \le 1$, $a_{ii} = 1$) denote similarity between bins i and j. Larger the a_{ij} the more the similarity between bins i and j. The two distributions can also be normalised so that $0 \le x_i, y_i \le 1$ and $\sum_i x_i = \sum_i y_i = 1$. Note that d_{hist} can be a positive semi-definite form even when $\mathbf{A}$ is an indefinite matrix because of the normalization constraints on the distributions. We assume that the above equations indeed define a positive semi-definite form. It can be shown that under certain conditions on the a_{ij}'s, it always has this property [5]. Besides, in any specific application or choice of matrix $\mathbf{A}$, this can be easily tested numerically.

In our case d_{hist} represents a generalised quadratic histogram distance measure for color matching. Typically, 256 colors are adequate to capture the color distributions of most natural scenes. In contrast with direct Euclidean distance, the general quadratic form allows for similarity matching between different colors (represented by the histogram bins). Each entry a_{ij} in the "similarity matrix" $\mathbf{A}$ attempts to capture the perceptual similarity between the colors represented by bins i and j. This method of comparing histograms is more sophisticated than current popular methods such as that of Swain [14].

However, as has already been noted, the histogram quadratic form measure is computationally intensive and it operates on high dimensional features. The problem at hand is to define a considerably less expensive measure on considerably lower dimensional features. We show that, under fairly general conditions, the cheaper distance, call it d_k for now, can be bounded by d_{hist} in the form $\lambda d_k \le d_{\text{hist}}$, for some positive constant λ. Thus, in order to retrieve images satisfying $d_{\text{hist}} \le \epsilon$, the inequality $d_k \le \epsilon/\lambda$ can be used to retrieve images quickly and without misses. The

expensive measure d_{hist} will then have to be applied only to the filtered set of images.

If the similarity information is ignored, then the problem is a straightforward distribution matching problem. Swain [14] treats his color matching problem in this way, and uses L_1 distance as a measure of the distance between two histograms. Other obvious candidates are the "relative entropy" or "Kullback-Leibler" distance [2] or the distance implied by the standard chi-squared statistical test [11]. Ioka [8] used the histogram quadratic form distance, but did not use the cheaper distance as a lower bound to efficiently filter out unwanted records.

3. HISTOGRAM DISTANCE

Given the N-dimensional normalised distributions $\mathbf{x}$ and $\mathbf{y}$, if $\mathbf{z} = \mathbf{x} - \mathbf{y}$, then $-1 \leq z_i \leq 1$, $\sum_i z_i = 0$, and $d_{hist}(\mathbf{x}, \mathbf{y}) = \mathbf{z}^T \mathbf{A} \mathbf{z}$. We call the set of such $\mathbf{z}$'s the histogram space (though it is not technically a linear space). Without loss of generality, $\mathbf{A}$ can be assumed to be a symmetric matrix.

An example (from [4]) will illustrate why similarity weighted comparison between color histograms leads to perceptually desirable results in contrast with comparisons based on direct Euclidean distance between the distributions. For simplicity, consider a histogram distribution of three colors, say red, orange and blue, with

$$\mathbf{A}_{\text{red,orange,blue}} = \begin{bmatrix} 1.0 & 0.9 & 0.0 \\ 0.9 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

where red and orange are considered highly similar. Consider a pure red image, $\mathbf{x} = [1.0,\ 0.0,\ 0.0]^T$, and a pure orange image, $\mathbf{y} = [0.0,\ 1.0,\ 0.0]^T$. The histogram distance of equation (1) is 0.2. This low distance reflects the perceptual similarity of the two images although their distributions populate distinct bins of the histogram so that their Euclidean distance is 1.0.

It can be shown that for certain choices of $\mathbf{A}$, d_{hist} will be non-negative on the histogram space [5]. With $d_{max} = \max_{i,j}(d_{ij})$ one such choice is, $a_{ij} = (1 - d_{ij}/d_{max})$. Other choices of similarity metrics that capture perceptual similarity of colors and make $d_{hist} \geq 0$ are valid also. We have used the above value in experiments on real images.

Now, d_{hist} can be written in a different form by eliminating the constraint $\sum_i z_i = 0$.

$$d_{hist} = \mathbf{z}^T \mathbf{A} \mathbf{z} = \begin{bmatrix} \tilde{\mathbf{z}}^T & z_N \end{bmatrix} \begin{bmatrix} \mathbf{A}_{N-1} & \mathbf{a}_{*N} \\ \mathbf{a}_{*N}^T & a_{NN} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{z}} \\ z_N \end{bmatrix},$$

where $\tilde{\mathbf{z}}^T = [z_1 \cdots z_{N-1}]$ and $\mathbf{A}$ has been decomposed into its top left $(N-1) \times (N-1)$ component and the rest ($\mathbf{a}_{*N}$ being the Nth column of $\mathbf{A}$ less the last entry a_{NN}). Applying $\sum_i z_i = 0$ to this, we get

$$d_{hist} = \tilde{\mathbf{z}}^T \left[\mathbf{A}_{N-1} - \mathbf{a}_{*N}\mathbf{1}^T - \mathbf{1}\mathbf{a}_{*N}^T + a_{NN}\mathbf{1}\mathbf{1}^T\right] \tilde{\mathbf{z}} \qquad (2)$$

which is written as $d_{hist} = \mathbf{z}^T \mathbf{A} \mathbf{z} = \tilde{\mathbf{z}}^T \tilde{\mathbf{A}} \tilde{\mathbf{z}}$, where

$$\tilde{\mathbf{A}} = \left[\mathbf{A}_{N-1} - \mathbf{a}_{*N}\mathbf{1}^T - \mathbf{1}\mathbf{a}_{*N}^T + a_{NN}\mathbf{1}\mathbf{1}^T\right] \qquad (3)$$

is an $(N-1) \times (N-1)$ modified similarity matrix, and $\mathbf{1}$ is a vector of $N-1$ ones.

4. AVERAGE COLOR DISTANCE

We now present a particularly intuitive distance measure, called the average color distance (the distance between average colors), as a first instance of a cheaper measure.

Given that each bin of a color histogram represents a three-dimensional color vector in a suitably defined color space, the average color of an image histogram is defined to be the weighted average color corresponding to the normalised color histogram distribution. Specifically, let $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2\ \cdots\ \mathbf{c}_N]$ be a $3 \times N$ matrix whose ith column is the color $\mathbf{c}_i = [\alpha_i\ \beta_i\ \gamma_i]^T$, where α, β and γ represent the magnitudes along the three color dimensions (R,G,B, for instance). Given two N-dimensional color histograms, $\mathbf{x}$ and $\mathbf{y}$, the 3×1 average color vector for each is

$$\mathbf{x}_{avg} = \mathbf{C}\mathbf{x}, \quad \mathbf{y}_{avg} = \mathbf{C}\mathbf{y}.$$

The average color distance is defined by

$$d_{avg} = (\mathbf{x}_{avg} - \mathbf{y}_{avg})^T (\mathbf{x}_{avg} - \mathbf{y}_{avg}) = \mathbf{z}^T \mathbf{C}^T \mathbf{C} \mathbf{z}.$$

With $\mathbf{W} = \mathbf{C}^T\mathbf{C}$, $d_{avg} = \mathbf{z}^T \mathbf{W} \mathbf{z} = \tilde{\mathbf{z}}^T \tilde{\mathbf{W}} \tilde{\mathbf{z}}$, where $\tilde{\mathbf{z}}$ is as defined in Section 3 and $\tilde{\mathbf{W}}$ is defined in terms of $\mathbf{W}$ similar to $\tilde{\mathbf{A}}$ of equation (3). Clearly by construction $\mathbf{W}$ and $\tilde{\mathbf{W}}$ are PSD (but *not* PD since their rank is at most three).

Note that d_{avg} is defined over 3-dimensional features, $\mathbf{x}_{avg}$ and $\mathbf{y}_{avg}$, in contrast with N-dimensional (typically 256 or 64) features for d_{hist}. Also, the average color can be precomputed (at database population/compilation time) and then it can be organized into an indexable (R^*-tree like) structure. Furthermore, d_{hist} can be bounded from below by a simple function of d_{avg} which ensures that indexing on d_{avg} will be without any false misses.

4.1. Bound between d_{hist} and d_{avg}

If $\tilde{\mathbf{A}}$ is PD, then d_{hist} is just a norm on the histogram space, which is a subset of a linear space. The matrix $\mathbf{C}$ maps this vector space into Euclidean 3-space with the standard norm. Thus, d_{avg} is just the Euclidean norm on the image of the histogram space under the linear map induced by $\mathbf{C}$. In this context, the induced linear map is a bounded linear operator. Consequently,

$$d_{avg} \leq \|\mathbf{C}\|^2 d_{hist},$$

where $\|\mathbf{C}\|$ is the norm of this linear operator. In other words, the existence of the required inequality is a consequence of a general theorem [6, pg. 57] about finite dimensional normed vector spaces and bounded linear operators. If, as in our case, neither $\tilde{\mathbf{A}}$ nor $\tilde{\mathbf{W}}$ are PD but only PSD then the above results do not apply directly. In the following we extend the result to this case and give a simple method for computing the multiplier in the inequality using standard results in constrained optimisation.

Theorem 1 *With d_{hist} and d_{avg} defined as above, if $\tilde{\mathbf{A}}$ is positive semi-definite, then for all vectors $\mathbf{x}$ and $\mathbf{y}$, $d_{hist} \geq \lambda_1 d_{avg}$, where λ_1 is the minimum eigenvalue of the generalised eigenvalue problem $\tilde{\mathbf{A}}\tilde{\mathbf{z}} = \lambda \tilde{\mathbf{W}} \tilde{\mathbf{z}}$.*

Proof: It will be shown that $\tilde{\mathbf{z}}^T \tilde{\mathbf{A}} \tilde{\mathbf{z}} \geq \lambda_1 \tilde{\mathbf{z}}^T \tilde{\mathbf{W}} \tilde{\mathbf{z}}$. Let $\eta > 0$. Because $\tilde{\mathbf{A}}$ is PSD there is a unique solution to the constrained minimisation problem

$$\min_{\tilde{\mathbf{z}}:\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}}=\eta} \tilde{\mathbf{z}}^T \tilde{\mathbf{A}} \tilde{\mathbf{z}}. \qquad (4)$$

This solution occurs at a value of $\tilde{\mathbf{z}}$ where the function $\tilde{\mathbf{z}}^T \tilde{\mathbf{A}} \tilde{\mathbf{z}} - \lambda\left(\tilde{\mathbf{z}}^T \tilde{\mathbf{W}} \tilde{\mathbf{z}} - \eta\right)$ has a critical value [9, Section 10.3]. These critical values (in the variables $\tilde{\mathbf{z}}$ and λ) occur when $\tilde{\mathbf{A}}\tilde{\mathbf{z}} = \lambda \tilde{\mathbf{W}} \tilde{\mathbf{z}}$ and $\tilde{\mathbf{z}}^T \tilde{\mathbf{W}} \tilde{\mathbf{z}} = \eta$, and these are regular points

because $\eta > 0$. Let $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{N-1}$, be the generalized eigenvalue solutions of the first equation and $\tilde{\mathbf{z}}_i$ be any generalized eigenvector also satisfying the second condition. Because the minimum in (4) exists, it must be the case that for some i, $1 \leq i \leq N-1$, $\tilde{\mathbf{z}}^T\tilde{\mathbf{A}}\tilde{\mathbf{z}} \geq \tilde{\mathbf{z}}_i^T\tilde{\mathbf{A}}\tilde{\mathbf{z}}_i$, for all $\tilde{\mathbf{z}}$ such that $\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}} = \eta$. But the right hand side of this equation is just $\lambda_i\eta$. Hence this minimum occurs for $i = 1$, the minimum eigenvalue.

We have shown that

$$\tilde{\mathbf{z}}^T\tilde{\mathbf{A}}\tilde{\mathbf{z}} \geq \lambda_1\eta = \lambda_1\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}},$$

for $\tilde{\mathbf{z}}$ such that $\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}} = \eta$, with $\eta > 0$ being completely arbitrary. If $\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\tilde{\mathbf{z}} = 0$, then this inequality holds *a priori* because $\tilde{\mathbf{A}}$ is PSD. Consequently, $d_{hist} \geq \lambda_1 d_{avg}$, as claimed. □

Note that λ_1 does not depend on the data, that is the image histograms, but only on the similarity matrix $\mathbf{A}$ and on the definitions of the colors in the matrix $\mathbf{C}$. In our experiments, we used the IMSL routine [7, pg. 454], *DGVLRG*, to compute λ_1. This routine does not require $\tilde{\mathbf{W}}$ or $\tilde{\mathbf{A}}$ to be positive definite.

Because both $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{W}}$ are PSD, $\lambda_1 \geq 0$. It is indeed possible that $\lambda_1 = 0$, even though this is of no use to us. This will occur when the null space of $\tilde{\mathbf{A}}$ is not contained in the null space of $\tilde{\mathbf{W}}$. This case is ignored throughout the rest of the paper. Furthermore, for $\lambda_1 > 0$, $\mathbf{C}$ (and so $\tilde{\mathbf{W}}$) could be normalized so that $\lambda_1 = 1$.

Since $d_{hist} \geq \lambda d_{avg}$, for any *range query* of the form $d_{hist} \leq \epsilon$, images retrieved using the filter $d_{avg} \leq \epsilon/\lambda$ are guaranteed to be a super set of the complete target set. If the filtering returns a relatively small fraction of the entire database, then the expensive d_{hist} needs to be computed only for that small fraction and not for the whole database.

If the 3-dimensional average color for each image, $(\mathbf{Cx})$, is precomputed and stored, then at query time the average color distance requires only three multiplications and subtractions, and two additions, i.e., is independent of N, the number of histogram bins.

5. EXPERIMENTAL RESULTS WITH d_{avg}

The quadratic distance, d_{hist}, has been in use in the QBIC system [12] for retrieval of images based on similarity of color histograms. Experiments with a variety of color spaces found that the best performance (using human judgement of the similarity between the query and database images) was obtained using a variant of the Munsell color space. This is a 3-dimensional space with the property that the Euclidean distance between two color vectors corresponds over a large part of the space to the perceptual difference between the colors. A transformation from the (R, G, B) space, in which the original images are defined, to the Munsell space, described in [10], was used. The histograms used are 256 dimensional.

We performed simulations to evaluate the effectiveness of filtering with d_{avg} on a database of 917 assorted natural images in the QBIC database. The relative retrieval efficiency between two methods is reported: (i) simple sequential evaluation of d_{hist}, for all database vectors, (referred to as *'naive'*), and (ii) filtering using d_{avg} followed by evaluation of d_{hist} only on those records that pass through the filter (referred to as *'filtered'*). Indexing methods (such as R^*-trees) were not used because the focus was on the gains from the filtering step. The sample queries involved matching each histogram record against the remaining records.

In the ideal case, the filtering step would filter out only exactly those records for which d_{hist} was in the desired range. It is guaranteed that filtering will include all records for which this is true, but some "false alarms" will also be retrieved. A was defined with a_{ij} as in section 3, guaranteeing it to be non-negative definite.

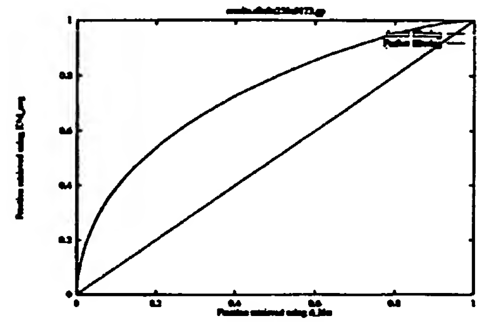


Figure 1: d_{hist} vs. d_{avg} retrieval.

Figure 1 shows the extent of the false alarms generated by d_{avg} for the test database. For instance, if the tolerance is set so that the top 10% of the database would have been retrieved by direct search using d_{hist}, then the filtering with d_{avg} retrieved approximately 40% of the database. Even this filtering results in a considerable saving because d_{hist} is applied only to the filtered 40% of the records and not to the complete 100%.

6. CHEAPER SIMILARITY MEASURES

The function d_{avg} is not the only distance that leads to low-dimensionality features and filtering of the database without any misses. It just happens to be one that results in an intuitively appealing feature, the average color. No particular assumptions were made on the matrix $\mathbf{C}$. Therefore, the question to ask is if the notion of a cheaper distance measure can be generalized.

Given a particular number, k, of feature dimensions, what is the optimum k-dimensional index and the corresponding distance? A requirement is that the distance measure should be based on the *precomputed* k-dimensional features like the average color. Similar to the definition of average color, let $\mathbf{x}_k = \mathbf{U}_k\tilde{\mathbf{x}}$ be a k-dimensional feature defined for the $(N-1)$-dimensional normalized and reduced histogram $\tilde{\mathbf{x}}$, where $\mathbf{U}_k$ is a $k \times (N-1)$ matrix. The k-distance (like d_{avg}) between two histograms $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ is given by:

$$d_k = d_k(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = (\mathbf{x}_k - \mathbf{y}_k)^T(\mathbf{x}_k - \mathbf{y}_k) = \tilde{\mathbf{z}}^T\mathbf{U}_k^T\mathbf{U}_k\tilde{\mathbf{z}} = \tilde{\mathbf{z}}^T\tilde{\mathbf{A}}_k\tilde{\mathbf{z}},$$

where $\tilde{\mathbf{A}}_k = \mathbf{U}_k^T\mathbf{U}_k$, is an at most k-rank PSD matrix.

The problem of determining the optimal d_k for a fixed k can be formalized as finding the rank k matrix $\tilde{\mathbf{A}}_k$ at which the extremum

$$\inf_{\tilde{\mathbf{A}}_k} \sup_{\tilde{\mathbf{z}}} \tilde{\mathbf{z}}^T(\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k)\tilde{\mathbf{z}} \qquad (5)$$

is attained, subject to (i) $(\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k)$ being PSD, and (ii) $\sum_i |\tilde{z}_i| \leq 1$. In words, we want to find that k-rank approximation to $\tilde{\mathbf{A}}$ such that the maximum difference (over $\tilde{\mathbf{z}}$)

between the true distance and the lower dimensional distance is a minimum for the given k. Also, because we want $d_{hist} \geq d_k$ to avoid false misses (with $\lambda = 1$), $(\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k)$ should be PSD.

The Singular Value Decomposition (SVD) (see, for example, [3]) of $\tilde{\mathbf{A}}$ provides a constructive answer to the above problem. Let $\tilde{\mathbf{A}} = \mathbf{V}^T\mathbf{\Sigma}\mathbf{V}$ be the SVD of $\tilde{\mathbf{A}}$, where $\mathbf{V}$ is an orthogonal matrix and $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_{N-1})$ with $\sigma_1 \geq \cdots \geq \sigma_k \geq \cdots \geq \sigma_{N-1} \geq 0$. Then, a solution to the problem in (5) is given by

$$\tilde{\mathbf{A}}_k = \mathbf{V}_k^T\mathbf{\Sigma}_k\mathbf{V}_k, \tag{6}$$

where $\mathbf{\Sigma}_k$ is the $k \times k$ diagonal matrix $\mathrm{diag}(\sigma_1, \ldots, \sigma_k)$, and $\mathbf{V}_k$ is the matrix whose rows are the first k rows of $\mathbf{V}$, i.e., corresponding to the first k singular values.

To see this, suppose for a moment that the singular values σ_j, $j = 1, \ldots, N-1$ are *not* ordered. The results in [3] show that an optimal k-rank approximation $\tilde{\mathbf{A}}_k$ to a matrix $\tilde{\mathbf{A}}$ must have the property that $\mathbf{V}\tilde{\mathbf{A}}_k\mathbf{V}^T$ is diagonal and have entries which come from the diagonal matrix $\mathbf{\Sigma}$, i.e., the singular values of $\tilde{\mathbf{A}}$. Consequently, the extremum in (5) can only be attained if $\tilde{\mathbf{A}}_k$ has the form given in equation (6). Now observe that under this assumption

$$d_{diff} = \tilde{\mathbf{z}}^T(\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k)\tilde{\mathbf{z}} = (\mathbf{V}\tilde{\mathbf{z}})^T(\mathbf{\Sigma} - \mathbf{\Sigma}'_k)(\mathbf{V}\tilde{\mathbf{z}}),$$

where $\mathbf{\Sigma}'_k = \begin{pmatrix} \mathbf{\Sigma}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$. From this we easily see that $(\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k)$ is PSD. Furthermore, the level set $d_{diff} = 1$ defines a hyperellipsoid in $N-1-k$ dimensions whose principal axes are the square-roots of the reciprocals of $\sigma_{k+1}, \ldots, \sigma_{N-1}$ (without loss of generality, we assume that these sigmas are non-zero else the subspace can be further reduced in dimension). The constraint $\sum_i |\tilde{z}_i| \leq 1$ defines a polytope with corners on each of the $\tilde{z}_i$ axes. The maximum of d_{diff} over such $\tilde{\mathbf{z}}$ will lie on one of the corners. The value of the maximum will be determined by the smallest principal axis of the ellipsoid, namely $\min_{k+1 \leq j \leq N-1}\{1/\sqrt{\sigma_j}\}$. The minimum, over all possible orderings of the σ's, of this maximum occurs when the smallest axis is largest. That is, the desired extremum occurs when $\sigma_1 \geq \cdots \geq \sigma_k \geq \cdots \geq \sigma_{N-1}$, as claimed above. Thus, $\tilde{\mathbf{A}}_k$ of (6) is indeed the solution to (5).

6.1. k-dimensional Index and d_k

The above result creates a method for constructing a k-dimensional index (or feature) for similarity search of N-dimensional histograms. In the spirit of the average color (a 3-dimensional feature or index), a *k-index* for an N-dimensional histogram, $\mathbf{x}$, can be defined as: $\mathbf{x}_k = \sqrt{\mathbf{\Sigma}_k}\mathbf{V}_k\tilde{\mathbf{z}}$. In practice, $k \ll N$ can be chosen arbitrarily though its choice should depend on the distribution of the significant singular values σ_i. For each image, $\mathbf{x}_k$ can be pre-computed and stored in the database. Just as with average color, the low-dimensional features $\mathbf{x}_k$ can be organized in some database structure to optimize the search at query time. In any case, d_k becomes a filter for d_{hist}.

Furthermore, in a large database, the cost of retrieving large records can overshadow even the cost of the full distance computation. On the other hand, for low dimensional searches, even sequential search accesses small records. Consequently, this filtering method can reduce computation costs both in disk access and distance computation. The only assumption here is that $\tilde{\mathbf{A}}$ has only a few significant singular values, so that the number of false alarms will be relatively small. As is shown in the next section, this indeed is the case in the $\tilde{\mathbf{A}}$'s that we use in our experiments.

7. EXPERIMENTAL RESULTS WITH d_k

We present experimental results for the selectivity of the filtering step using d_k for various values of k. The range of singular values of the matrix $\tilde{\mathbf{A}}$ which we used is 150.78 to 0.03. The first 12 singular values (in descending order) of the 255 × 255 matrix are 150.78, 20.32, 9.81, 7.07, 3.73, 2.45, 1.96, 1.54, 1.31, 1.15, 1.01. All other singular values are less than 1.0.

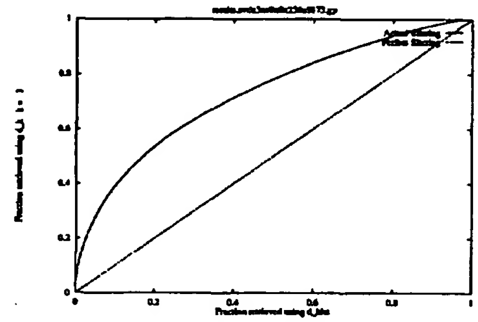


Figure 2: d_{hist} vs. d_k retrieval for $k = 3$.

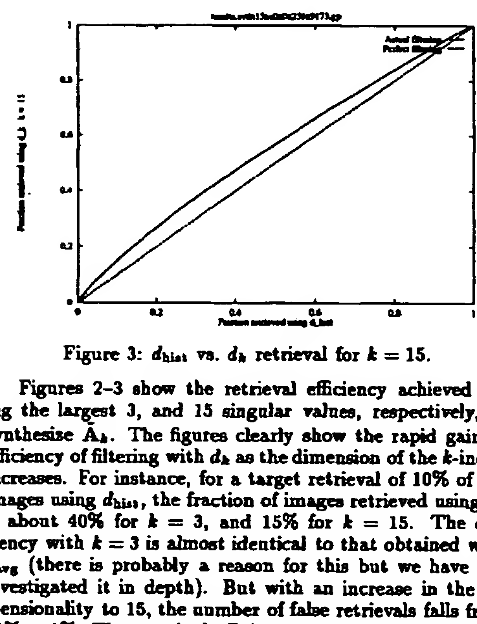


Figure 3: d_{hist} vs. d_k retrieval for $k = 15$.

Figures 2–3 show the retrieval efficiency achieved using the largest 3, and 15 singular values, respectively, to synthesize $\tilde{\mathbf{A}}_k$. The figures clearly show the rapid gain in efficiency of filtering with d_k as the dimension of the k-index increases. For instance, for a target retrieval of 10% of the images using d_{hist}, the fraction of images retrieved using d_k is about 40% for $k = 3$, and 15% for $k = 15$. The efficiency with $k = 3$ is almost identical to that obtained with d_{avg} (there is probably a reason for this but we have not investigated it in depth). But with an increase in the dimensionality to 15, the number of false retrievals falls from 30% to 8%. The marginal efficiency increase tapers off with further increase in k.

Figure 4 shows the ratio of retrieval using d_k to retrieval using d_{hist}, for 10% retrieval with d_{hist}, as a function of k.

The asymptotic behavior of the ratio (efficiency) is evident from this figure.

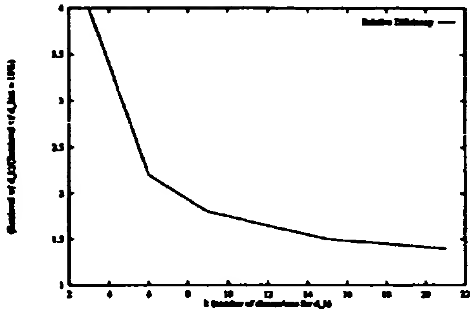


Figure 4: Retrieval ratio with d_k to d_{hist} vs. k (10% of database retrieved by d_{hist}).

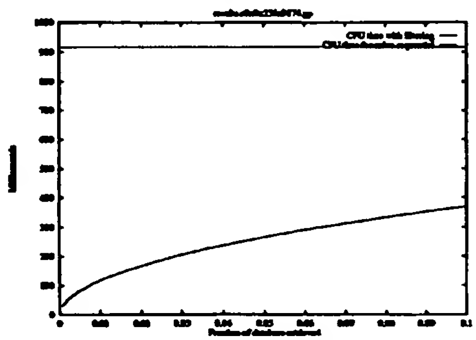


Figure 5: CPU time for *naive* retrieval vs. filtered retrieval with d_{avg}.

In order to illustrate the efficiency gained in CPU time only, figure 5 shows the relative CPU times (time taken for distance computation only), for up to 10% retrieval, between d_{hist} computation over the entire database, and d_{hist} computation over only the part of the database filtered using d_{avg}. Similarly, figure 6, shows the CPU times for filtering with d_k with $k = 15$. The CPU time for a 256-bin d_{hist} was about 1*ms.* on an RS6000/350. It is clear from the comparisons that even for retrievals as high as 10% of the database, d_{avg} filtering results in almost 1/3rd the CPU time, and d_k ($k = 15$) results in about 1/6th the CPU time of naive d_{hist} retrieval. It is assumed in this comparison that both for direct d_{hist} querying and filtered querying, all the database records are read into memory before the distance computations. Given that the focus of this paper is on the filtering efficiency, a detailed timing comparison with realistic models and simulations of efficient indexing, combined disk access and CPU times has been left for another time.

8. CONCLUSIONS

We presented a method for efficient retrieval of images based on similarity of color histograms. Clearly, the method may also be useful for any other application involving a quadratic distance measure. The results indicate that

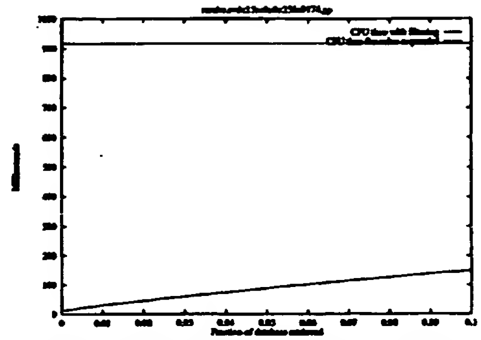


Figure 6: CPU time for *naive* retrieval vs. filtered retrieval with d_k for $k = 15$.

the simple filtering method described here easily outperforms the naive histogram comparison method. Using low-dimensional indexing techniques (e.g., k-index) would certainly make our method more attractive for very large database applications.

9. REFERENCES

[1] R. Agrawal et al. Efficient similarity search in sequence databases. In *4th Intl. Conf. on Foundations of Data Organization and Algorithms*, Chicago, IL, 1993.

[2] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley Series in Telecommunications. John Wiley & Sons, Inc., 1991.

[3] P. Dewilde and Ed. F. Deprettere. SVD, an intro. In Deprettere Ed. F, editor, *SVD and Sig. Proc. : Algs., Apps. and Archs.*, pages 3–42. Elsevier Sc. Pub. Co., Inc., NY.

[4] W. Equitz et al. Retrieving images from a database using color — algorithms from the QBIC system. Res. Rep. RJ 9377, IBM Almaden Res. Ctr., San Jose, CA, June 1993.

[5] Tomas Feder. On a quadratic form for points in space. IBM Internal Note, 1993.

[6] Gene H. Golub and Charles F. van Loan. *Matrix Computations.* The Johns Hopkins University Press, 1989.

[7] IMSL, Inc. *User's Manual MATH/LIBRARY.* IMSL, Inc., TX, 1991.

[8] Mikihiro Ioka. A method of defining the similarity of images on the basis of color information. Technical report RT-0030, IBM Tokyo Research Lab, 1989.

[9] David G. Luenberger. *Linear and Nonlinear Optimization.* Addison-Wesley, MA, 1984.

[10] M. Miyahara and Y. Yoshida. Math. transform of (R,G,B) color data to Munsell (H,V,C) color data. In *Vis. Comm. and Image Proc.*, volume 1001, pages 650–657. SPIE, 1988.

[11] A. Mood, F. Graybill, and D. Boes. *Introduction to the Theory of Statistics.* McGraw-Hill series in probability and statistics. McGraw-Hill, third edition, 1963.

[12] W. Niblack et al. The QBIC project: Querying Ims. by content using color, texture, and shape. In *SPIE 1908, Storage and Retrieval for Image and Video Dbases*, Feb. 1993.

[13] M. Otterman. Approx. matching with hi-dimensionality R-trees. M. Sc. scholarly paper, Dept. of CS, Univ. of MD, College Park, MD, supervised by C. Faloutsos., 1992.

[14] Michael J. Swain and Dana H. Ballard. Color indexing. *Intl. J. of Computer Vision*, 7(1):11–32, 1991.

Histogram refinement for content-based image retr

Pass, G. Zabih, R.
Dept. of Comput. Sci., Cornell Univ., Ithaca, NY , USA;

This paper appears in: **Applications of Computer Vision, 1996. WACV '96 Proceedings 3rd IEEE Workshop on**

Meeting Date: 12/02/1996 - 12/04/1996
Publication Date: 2-4 Dec. 1996
Location: Sarasota, FL USA
On page(s): 96 - 102
Reference Cited: 12
Number of Pages: xi+292
Inspec Accession Number: 5485094

Abstract:
Color histograms are widely used for content-based image retrieval. Their adv are efficiency, and insensitivity to small changes in camera viewpoint. Howeve histogram is a coarse characterization of an image, and so images with very d appearances can have similar histograms. We describe a technique for compa called histogram refinement, which imposes additional constraints on histogra matching. Histogram refinement splits the pixels in a given bucket into severa based upon some local property. Within a given bucket, only pixels in the sam compared. We describe a split histogram called a color coherence vector (CCV partitions each histogram bucket based on spatial coherence. CCVs can be com over 5 images per second on a standard workstation. A database with 15,000 be queried using CCVs in under 2 seconds. We demonstrate that histogram re can be used to distinguish images whose color histograms are indistinguishab

Index Terms:
image processing visual databases color coherence vector content-based image retr histogram refinement spatial coherence

Documents that cite this document
Select link to view other documents in the database that cite this one.

Search Results [PDF FULL-TEXT 1572 KB] DOWNLOAD CITATION

Histogram Refinement for Content-Based Image Retrieval

Greg Pass Ramin Zabih*

Computer Science Department
Cornell University
Ithaca, NY 14853
gregpass,rdz@cs.cornell.edu
http://www.cs.cornell.edu/home/rdz/refinement.html

Abstract

Color histograms are widely used for content-based image retrieval. Their advantages are efficiency, and insensitivity to small changes in camera viewpoint. However, a histogram is a coarse characterization of an image, and so images with very different appearances can have similar histograms. We describe a technique for comparing images called histogram refinement, *which imposes additional constraints on histogram based matching. Histogram refinement splits the pixels in a given bucket into several classes, based upon some local property. Within a given bucket, only pixels in the same class are compared. We describe a split histogram called a* color coherence vector *(CCV), which partitions each histogram bucket based on spatial coherence. CCV's can be computed at over 5 images per second on a standard workstation. A database with 15,000 images can be queried using CCV's in under 2 seconds. We demonstrate that histogram refinement can be used to distinguish images whose color histograms are indistinguishable.*

1 Introduction

Many applications require methods for comparing images based on their overall appearance. Color histograms are a popular solution to this problem, and are used in systems like QBIC [2] and Chabot [6]. Color histograms are computationally efficient, and generally insensitive to small changes in camera position. However, a color histogram provides only a very coarse characterization of an image; images with similar histograms can have dramatically different appearances. For example, the images shown in figure 1 have similar color histograms.

In this paper we describe a method which imposes additional constraints on histogram based matching. In *histogram refinement*, the pixels within a given bucket are split into classes based upon some local property. Split histograms are compared on a bucket by bucket basis, similar to standard histogram matching. Within a given bucket, only pixels with the same property are compared. Two images with identical color histograms can have different split histograms; thus, split histograms create a finer distinction than color histograms. This is particularly important for large image databases, in which many images can have similar color histograms.

Figure 1: Two images with similar color histograms

We have experimented with a split histogram called a *color coherence vector* (CCV), which partitions pixels based upon their spatial coherence. A coherent pixel is part of some sizable contiguous region, while an incoherent pixel is not. While the two images shown in figure 1 have similar color histograms, their CCV's are very different.[1] For example, red pixels appear in both images in similar quantities. In the left image the red pixels (from the flowers) are widely scattered, while in the right image the red pixels (from the golfer's shirt) form a single coherent region.

We begin with a review of color histograms. In section 3 we describe histogram refinement, and present two examples that capture spatial information. Section 4 provides examples of refinement-based image queries and shows that they can give superior results to color histograms. We compare our work with some recent algorithms [5, 8, 9, 10] that also combine spatial information with color histograms.

2 Color Histograms

Color histograms are frequently used to compare images. Examples of their use in multimedia applications include scene break detection and querying a database of images [7, 6, 2]. Color histograms are popular because they are trivial to compute, and tend to

*To whom correspondence should be addressed

[1]The color images used in this paper can be found at http://www.cs.cornell.edu/home/rdz/refinement.html.

0-8186-7620-5/96 $5.00 © 1996 IEEE

be robust against small changes in camera viewpoint. For example, Swain and Ballard [12] describe the use of color histograms for identifying objects. Stricker and Swain [11] analyze the information capacity of color histograms.

We will assume that all images are scaled to contain the same number of pixels M. We discretize the colorspace of the image such that there are n distinct (discretized) colors. A color histogram H is a vector $\langle h_1, h_2, \ldots, h_n \rangle$, in which each bucket h_j contains the number of pixels of color j in the image. Typically images are represented in the RGB colorspace, with a few of the most significant bits per color channel.

For a given image I, the color histogram H_I is a compact summary of the image. A database of images can be queried to find the most similar image to I, and can return the image I' with the most similar color histogram $H_{I'}$. Color histograms are typically compared using the L_1-distance or the L_2-distance, although more complex measures have also been considered [4].

3 Histogram Refinement

In *histogram refinement* the pixels of a given bucket are subdivided into classes based on local features. There are many possible features, including texture, orientation, distance from the nearest edge, relative brightness, etc. Histogram refinement prevents pixels in the same bucket from matching each other if they do not fall into the same class. Pixels in the same class can be compared using any standard method for comparing histogram buckets (such as the L_1 distance). This allows fine distinctions that cannot be made with color histograms.

As a simple example of histogram refinement, consider a positional refinement where each pixel in a given color bucket is classified as either "in the center" of the image, or not. Specifically, the centermost 75% of the pixels are defined as the "center". This produces a split histogram in which the pixels of color buckets are loosely constrained by their location in the image. The resulting split histograms can be compared using the L_1 distance. We will call this simple form of histogram refinement *centering refinement*.

Color coherence vectors

CCV's are a more sophisticated form of histogram refinement, in which histogram buckets are partitioned based on spatial coherence. Our coherence measure classifies pixels as either coherent or incoherent. A coherent pixel is a part of a sizable contiguous region, while an incoherent pixel is not. A *color coherence vector* represents this classification for each color in the image.

The initial stage in computing a CCV is similar to the computation of a color histogram. We first blur the image slightly by replacing pixel values with the average value in a small local neighborhood (currently including the 8 adjacent pixels). We then discretize the colorspace, such that there are only n distinct colors in the image.

The next step is to classify the pixels within a given color bucket as either coherent or incoherent. A coherent pixel is part of a large group of pixels of the same color, while an incoherent pixel is not. We determine the pixel groups by computing connected components. A connected component C is a maximal set of pixels such that for any two pixels $p, p' \in C$, there is a path in C between p and p'. We compute connected components using 4-connected neighbors within a given discretized color bucket. We classify pixels as either coherent or incoherent depending on the size in pixels of its connected component. A pixel is coherent if the size of its connected component exceeds a fixed value τ; otherwise, the pixel is incoherent.

For a given discretized color, some of the pixels with that color will be coherent and some will be incoherent. Let us call the number of coherent pixels of the j'th discretized color α_j and the number of incoherent pixels β_j. Clearly, the total number of pixels with that color is $\alpha_j + \beta_j$, and so a color histogram would summarize an image as $\langle \alpha_1 + \beta_1, \ldots, \alpha_n + \beta_n \rangle$. Instead, for each color we compute the pair (α_j, β_j) which we will call the *coherence pair* for the j'th color. The *color coherence vector* for the image consists of $\langle (\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n) \rangle$. This is a vector of coherence pairs, one for each discretized color.

In our experiments, all images were scaled to contain $M = 38,976$ pixels, and we have used $\tau = 300$ pixels (so a region is classified as coherent if its area is about 1% of the image). With this value of τ, an average image in our database consists of approximately 75% coherent pixels, with a standard deviation of 11%.

Two images I, I' can be compared using their CCV's, for example by using the L_1 distance. Let the coherence pairs for the j'th color bucket be (α_j, β_j) in I and (α'_j, β'_j) in I'. Using the L_1 distance to compare CCV's, the j'th bucket's contribution to the distance between I and I' is

$$\Delta_{CCV} = \left|(\alpha_j - \alpha'_j)\right| + \left|(\beta_j - \beta'_j)\right|. \qquad (1)$$

Note that when using color histograms to compare I and I', the j'th bucket's contribution is

$$\Delta_{CH} = \left|(\alpha_j + \beta_j) - (\alpha'_j + \beta'_j)\right|. \qquad (2)$$

It follows that CCV's create a finer distinction than color histograms. A given color bucket j can contain the same number of pixels in I as in I', but these pixels may be entirely incoherent in I and entirely coherent in I' (i.e., $\alpha = \beta' = 0$). Formally, $\Delta_{CH} \leq \Delta_{CCV}$ follows from equations 1 and 2, and the fact that the L_1 distance obeys the triangle inequality.

4 Experimental Results

We have implemented histogram refinement, and have used it for image retrieval from a large database. Our database consists of 14,554 images, which are drawn from a variety of sources. Our largest sources include the 11,667 images used in Chabot [6], the 1,440 images used in QBIC [2], and a 1,005 image database available from Corel. In addition, we included a few groups of images in PhotoCD format. Finally, we have taken a number of MPEG videos from the Web and segmented them into scenes. We have added one or

two images from each scene to the database, totaling 349 images. The image database thus contains a wide variety of imagery.

We have compared our results with a number of color histogram variants. These include the L_1 and L_2 distances, with both 64 and 512 color buckets. We include a small amount of smoothing as it empirically improved performance. On our database, the L_1 distance with the 64-bucket RGB colorspace gave the best results, and is used as a benchmark.

Hand examination of our database revealed 75 pairs of images which contain different views of the same scene. Examples are shown in figures 2 and 3. One image is selected as a query image, and the other represents a "correct" answer. In each case, we have shown where the second image ranks, when similarity is computed using color histograms or using histogram refinement. Specifically, results are shown using CCV's, centering refinement, and a successive refinement technique described in section 6.1. The color images shown are available at http://www.cs.cornell.edu/home/rdz/refinement.html.

4.1 Centering refinement results

In 69 of the 75 cases, centering refinement produced better results, while in 4 cases it produced worse results (there were 2 cases where the ranks did not change). The average change in rank due to centering refinement was an improvement of 55 positions (this included all 75 cases). The average percentage change in rank was an improvement of 41%. In the 69 cases where centering refinement performed better than color histograms, the average improvement in rank was 60 positions, and the average percentage improvement was 49%. In the 4 cases where color histograms performed better than centering refinement, the average rank improvement was 10 positions. We have not yet analyzed these 4 cases to determine why centering refinement fails.

To analyze the statistical significance of this data, we formulate the null hypothesis H_0 which states that centering refinement is equally likely to cause a positive change in ranks (i.e., an improvement) or a negative change. We will discard the 2 ties to simplify the analysis. Under H_0, the expected number of positive changes is 36.5, with a standard deviation of $\sqrt{73}/2 \approx 4.27$. The actual number of positive changes is 69, which is more than 7.6 standard deviations greater than the number expected under H_0. We can therefore reject H_0 at any standard significance level (such as 99.9%).

4.2 CCV results

In 68 of the 75 cases, CCV's produced better results, while in 7 cases they produced worse results. The average change in rank due to CCV's was an improvement of 68 positions (note that this included the 7 cases where CCV's do worse). The average percentage change in rank was an improvement of 35%. In the 68 cases where CCV's performed better than color histograms, the average improvement in rank was 77 positions, and the average percentage improvement was 56%. In the 7 cases where color histograms performed better, the average improvement was 17 positions.

The null hypothesis H_0 states that CCV's are equally likely to cause a positive change in ranks (i.e., an improvement) or a negative change. Under H_0, the expected number of positive changes is 37.5, with a standard deviation of $\sqrt{75}/2 \approx 4.33$. The actual number of positive changes is 68, which is more than 7 standard deviations greater than the number expected under H_0. We can therefore reject H_0 at any standard significance level (such as 99.9%).

When CCV's produced worse results, it was always due to a change in overall image brightness (i.e., the two images were almost identical, except that one was brighter than the other). Because CCV's use discretized color buckets for segmentation, they are more sensitive to changes in overall image brightness than color histograms. We believe that this difficulty can be overcome by using a better colorspace than RGB, as we discuss in section 6.2.

4.3 Efficiency

We have experimented with a number of different techniques for histogram refinement. CCV's are the most computationally expensive method of these, and will be our focus in discussing efficiency.

There are two phases to the computation involved in querying an image database. First, when an image is inserted into the database, a CCV must be computed. Second, when the database is queried, some number of the most similar images must be retrieved. Most methods for content-based indexing include these distinct phases. For both color histograms and CCV's, these phases can be implemented in linear time with a single pass over the image.

We ran our experiments on a 50 MHz SPARCstation 20, and provide the results from color histogramming for comparison. Color histograms can be computed at 67 images per second, while CCV's can be computed at 5 images per second. Using color histograms, 21,940 comparisons can be performed per second, while with CCV's 7,746 can be performed per second. The images used for benchmarking are 232×168. Both implementations are preliminary, and the performance can definitely be improved.

5 Related Work

Our work has focused on the use of spatial information to refine color histograms. Recently, several authors have proposed algorithms for comparing images that combine spatial information with color histograms. Hsu *et al.* [5] attempts to capture the spatial arrangement of the different colors in the image, in order to perform more accurate content-based image retrieval. Rickman and Stonham [8] randomly sample the endpoints of small triangles and compare the distributions of these triplets. Smith and Chang [9] concentrate on queries that combine spatial information with color. Stricker and Dimai [10] divide the image into five partially overlapping regions and compute the first three moments of the color distributions in each image. We will discuss each approach in turn.

Hsu [5] begins by selecting a set of representative colors from the image. Next, the image is partitioned

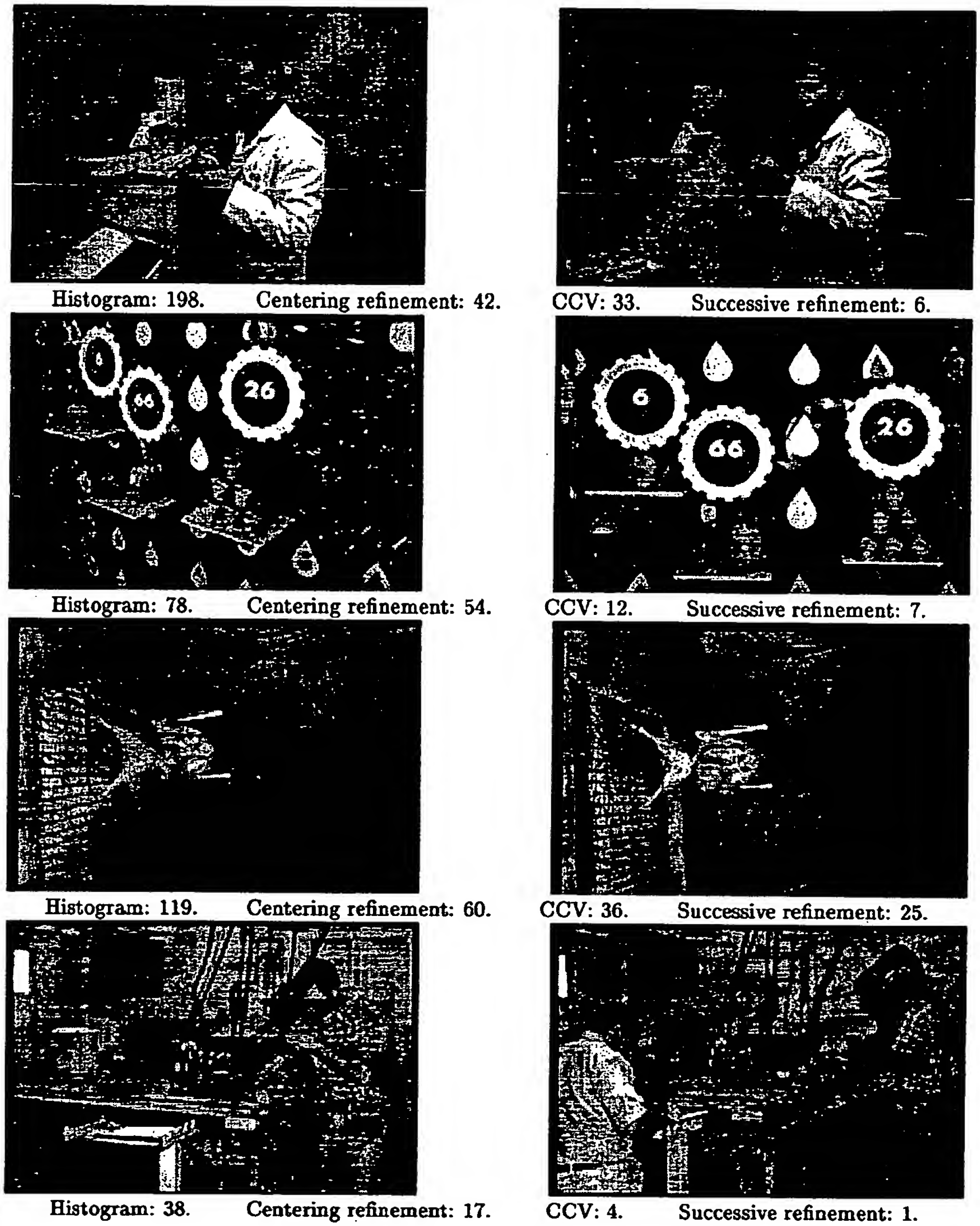
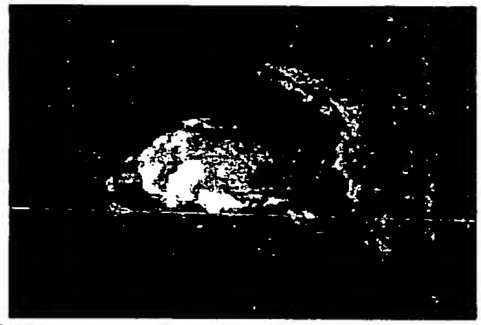
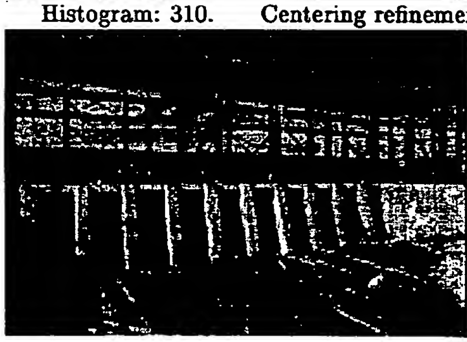
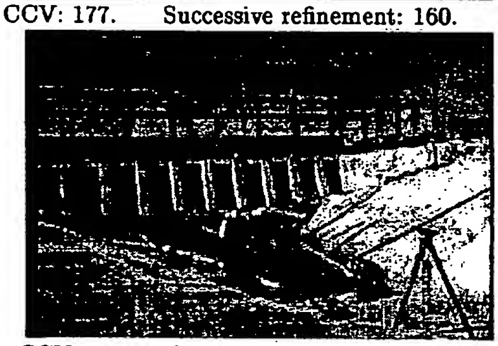


Figure 2: Example queries with their partner images, plus ranks under various methods. Lower ranks indicate better performance.

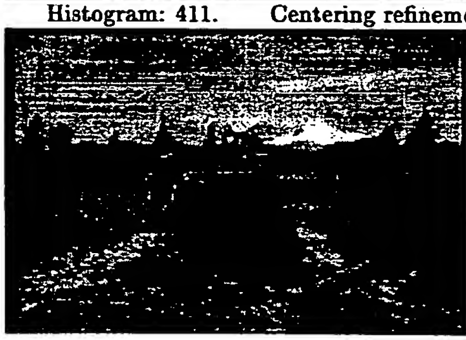
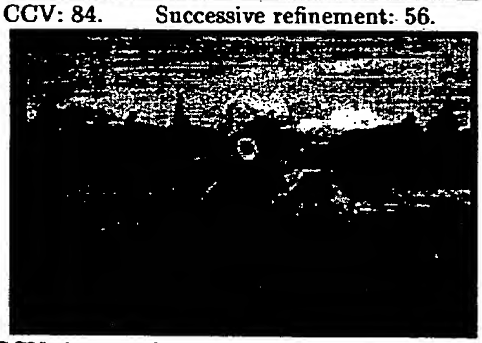



Histogram: 88. Centering refinement: 35. CCV: 20. Successive refinement: 13.


Histogram: 310. Centering refinement: 214. CCV: 177. Successive refinement: 160.

Histogram: 411. Centering refinement: 282. CCV: 84. Successive refinement: 56.

Histogram: 50. Centering refinement: 37. CCV: 27. Successive refinement: 22.

Figure 3: Additional example queries with ranks. Lower ranks indicate better performance.

into rectangular regions, where each region is predominantly a single color. The partitioning algorithm makes use of maximum entropy. The similarity between two images is the degree of overlap between regions of the same color. Hsu presents results from a database with 260 images, which show that their approach can give better results than color histograms.

While the authors do not report running times, it appears that Hsu's method requires substantially more computation than the approach we describe. A CCV can be computed in a single pass over the image, with a small number of operations per pixel. Hsu's partitioning algorithm in particular appears much more computationally intensive than our method. Hsu's approach can be extended to be independent of orientation and position, but the computation involved is quite substantial. In contrast, our method is naturally invariant to orientation and position.

Rickman and Stonham [8] randomly sample pixel triples arranged in an equilateral triangle with a fixed side length. They use 16 levels of color hue, with non-uniform quantization. Approximately a quarter of the pixels are selected for sampling, and their method stores 372 bits per image. They report results from a database of 100 images.

Smith and Chang's algorithm also partitions the image into regions, but their approach is more elaborate than Hsu's. They allow a region to contain multiple different colors, and permit a given pixel to belong to several different regions. Their computation makes use of histogram back-projection [12] to back-project sets of colors onto the image. They then identify color sets with large connected components.

Smith and Chang's image database contains 3,100 images. Again, running times are not reported, although their algorithm does speed up back-projection queries by pre-computing the back-projections of certain color sets. Their algorithm can also handle certain kinds of queries that our work does not address; for example, they can find all the images where the sun is setting in the upper left part of the image.

Stricker and Dimai [10] compute moments for each channel in the HSV colorspace, where pixels close to the border have less weight. They store 45 floating point numbers per image. Their distance measure for two regions is a weighted sum of the differences in each of the three moments. The distance measure for a pair of images is the sum of the distance between the center regions, plus (for each of the 4 side regions) the minimum distance of that region to the corresponding region in the other image, when rotated by 0, 90, 180 or 270 degrees. Because the regions overlap, their method is insensitive to small rotations or translations. Because they explicitly handle rotations of 0, 90, 180 or 270 degrees, their method is not affected by these particular rotations. Their database contains over 11,000 images, but the performance of their method is only illustrated on 3 example queries. Like Smith and Chang, their method is designed to handle certain kinds of more complex queries that we do not consider.

6 Extensions

There are a number of ways in which our histogram refinement could be extended and improved. One generalization is to further subdivide split histograms based on additional features; we refer to this process as *successive refinement*. Another extension centers on improving the choice of colorspace.

6.1 Successive refinement

In *successive refinement* the buckets in a split histogram are further subdivided based on additional features. Much as we distinguish between pixels of similar color by coherence, we can distinguish between pixels of similar coherence by some additional feature. We can apply this method repeatedly; each refinement imposes an additional constraint on what it means for two pixels to be similar.

We have implemented a simple successively refined histogram. A color histogram was first split with coherence constraints (to create a CCV). Successive refinement was enforced on both the coherent and incoherent pixels of the CCV. We used the centering refinement introduced in section 3. With successive refinement, pixels are divided into four classes based on coherence versus incoherence, and on whether or not they were in the centermost 75% of the image. The L_1 distance was used as a comparison measure. Examples of the successively refined histogram's performance are shown in figures 2 and 3. These preliminary results seem promising.

We have also investigated successive refinement based on intensity gradients. Again, the initial refinement was based on coherence, and the successive refinement was enforced identically on coherent and incoherent pixels. We have further classified pixels based on the gradient magnitude or on the gradient direction. The results we obtained are quite preliminary, but they seem to indicate a statistically significant improvement over CCV's.

The best system of constraints to impose on the image is an open issue. Any combination of features might give effective results, and there are many possible features to choose from. However, it is possible to take advantage of the temporal structure of a successively refined histogram. One feature might serve as a filter for another feature, by ensuring that the second feature is only computed on pixels which already possess the first feature.

For example, the perimeter-to-area ratio can be used to classify the relative shapes of color regions. If we used this ratio as an initial refinement on color histograms, incoherent pixels would result in statistical outliers, and thus give questionable results. This feature is better employed after the coherent pixels have been segregated. Refining a histogram not only makes finer distinctions between pixels, but functions as a statistical filter for successive refinements.

6.2 Choice of colorspace

Many researchers spend considerable effort on selecting a good set of colors. Hsu [5], for example, assumes that the colors in the center of the image are more important than those at the periphery, while Smith and Chang [9] use several different thresholds to

extract colors and regions. A wide variety of different colorspaces have also been investigated for content-based image retrieval, such as the opponent-axis colorspace [12] and the Munsell colorspace [2].

The choice of colorspace is a particularly significant issue for CCV's, since they use the discretized color buckets to segment the image. A perceptually uniform colorspace, such as CIE Lab, should result in better segmentations and improve the performance of CCV's. A related issue is the color constancy problem, which causes objects of the same color to appear rather differently depending upon the lighting conditions. The simplest effect of color constancy is a change in overall image brightness; this is responsible for the negative examples obtained in our experiments with CCV's. Standard histogramming methods are sensitive to image gain. More sophisticated methods, such as color ratio histograms [3] or the use of color moments [10], might alleviate this problem. These methods, like most proposed improvements to color histograms, can also be used in histogram refinement. For example, color moments could be computed separately for coherent and incoherent pixels.

7 Conclusions

We have described a method for imposing additional constraints on histogram based matching called histogram refinement. This idea can be extended by placing further constraints on the split histogram itself. Both histogram refinement and successive refinement are general methods for improving the performance of histogram based matching. If the initial histogram is a color histogram, and it is refined based on coherence, then the resulting split histogram is a CCV. But there is no requirement that this refinement be based on coherence, or even that the initial histogram be based on color.

Most research in content-based image retrieval has focused on query by example (where the system automatically finds images similar to an input image). However, other types of queries are also important. For example, it is often useful to search for images in which a subset of another image (e.g. a particular object) appears. This would be particularly useful for queries on a database of videos. One approach to this problem might be to generalize histogram backprojection [12] to separate pixels based on spatial coherence, or some other local property.

It is clear that larger and larger image databases will demand more complex similarity measures. This added time complexity can be offset by using efficient, coarse measures that prune the search space by removing images which are clearly not the desired answer. Measures which are less efficient but more effective can then be applied to the remaining images. Baker and Nayar [1] have begun to investigate similar ideas for pattern recognition problems. To effectively handle large image databases will require a balance between increasingly fine measures (such as histogram refinement) and efficient coarse measures.

Acknowledgments

We wish to thank Virginia Ogle for giving us access to the Chabot imagery, and Thorsten von Eicken for supplying additional images. Greg Pass has been supported by Cornell's Alumni-Sponsored Undergraduate Research Program. We also thank Vera Kettnaker and Justin Miller for helping produce the data.

References

[1] Simon Baker and Shree Nayar. Pattern rejection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 544–549, 1996.

[2] M. Flickner *et al.* Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23–32, September 1995.

[3] Brian V. Funt and Graham D. Finlayson. Color constant color indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):522–529, May 1995.

[4] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, July 1995.

[5] Wynne Hsu, T. S. Chua, and H. K. Pung. An integrated color-spatial approach to content-based image retrieval. In *ACM Multimedia Conference*, pages 305–313, 1995.

[6] Virginia Ogle and Michael Stonebraker. Chabot: Retrieval from a relational database of images. *IEEE Computer*, 28(9):40–48, September 1995.

[7] Alex Pentland, Rosalind Picard, and Stan Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, June 1996.

[8] Rick Rickman and John Stonham. Content-based image retrieval using color tuple histograms. *SPIE proceedings*, 2670:2–7, February 1996.

[9] John Smith and Shih-Fu Chang. Tools and techniques for color image retrieval. *SPIE proceedings*, 2670:1630–1639, February 1996.

[10] Markus Stricker and Alexander Dimai. Color indexing with weak spatial constraints. *SPIE proceedings*, 2670:29–40, February 1996.

[11] Markus Stricker and Michael Swain. The capacity of color histogram indexing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 704–708, 1994.

[12] Michael Swain and Dana Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.